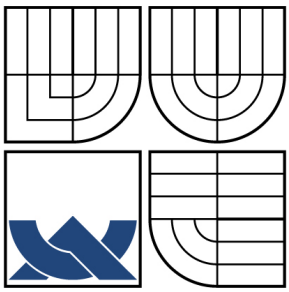
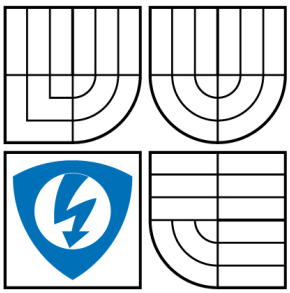


VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ

ÚSTAV MIKROELEKTRONIKY

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION

DEPARTMENT OF MICROELECTRONICS

ZAŘÍZENÍ PRO OVLÁDÁNÍ MIKROPOSUVU

MICROPOSITION CONTROL DEVICE

DIPLOMOVÁ PRÁCE

MASTER'S THESIS

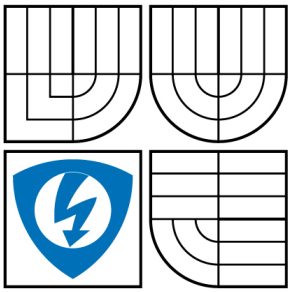
AUTOR PRÁCE — Bc. JAN CHMELÍK

AUTHOR

VEDOUCÍ PRÁCE — Ing. JAROSLAV KADLEC, Ph.D.

SUPERVISOR

BRNO 2008



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

Fakulta elektrotechniky a komunikačních technologií

Ústav mikroelektroniky

Diplomová práce

magisterský navazující studijní obor
Mikroelektronika

Student:	Chmelík Jan Bc.	***ID:***	21933
Ročník:	2	***Akademický rok:***	2007/2008

NÁZEV TÉMATU:

Zařízení pro ovládání mikroposuvu

POKYNY PRO VYPRACOVÁNÍ:

Navrhněte a realizujte zařízení umožňující ovládání mikroposuvu prostřednictvím mikrokontroléru. Zařízení má být schopno ovládat mikroposuv s nastavitelnou rychlostí posuvu a velikostí kroku, přednastavovat uživatelsky definovaný profil posouvání a komunikovat s osobním počítačem prostřednictvím sběrnice USB, případně RS232. Součástí tohoto projektu je programové vybavení mikrokontroléru i osobního počítače.

DOPORUČENÁ LITERATURA:

dle pokynů vedoucího práce

Termín zadání:	5.10.2007	***Termín odevzdání:***	26.5.2008

Vedoucí práce: Ing. Jaroslav Kadlec, Ph.D.

prof. Ing. Vladislav Musil, CSc.
předseda oborové rady

LICENČNÍ SMLOUVA
POSKYTOVANÁ K VÝKONU PRÁVA UŽÍT ŠKOLNÍ DÍLO

uzavřená mezi smluvními stranami:

1. Pan/paní

Jméno a příjmení: Bc. Jan Chmelík
Bytem: Štolcova 967/52B, 61800, Brno - Černovice
Narozen/a (datum a místo): 30.3.1982, Brno

(dále jen "autor")

a

2. Vysoké učení technické v Brně

Fakulta elektrotechniky a komunikačních technologií
se sídlem Údolní 244/53, 60200 Brno 2
jejímž jménem jedná na základě písemného pověření děkanem fakulty:
Ing. Edita Hejátková

(dále jen "nabyvatel")

Článek 1
Specifikace školního díla

1. Předmětem této smlouvy je vysokoškolská kvalifikační práce (VŠKP):
   - ☐ disertační práce
   - ☑ diplomová práce
   - ☐ bakalářská práce

   jiná práce, jejíž druh je specifikován jako .....

(dále jen VŠKP nebo dílo)

Název VŠKP: Zařízení pro ovládání mikroposuvu
Vedoucí/školitel VŠKP: Ing. Jaroslav Kadlec, Ph.D.
Ústav: Ústav mikroelektroniky
Datum obhajoby VŠKP: .....

VŠKP odevzdal autor nabyvateli v:

- ☐ tištěné formě - počet exemplářů .....
- ☐ elektronické formě - počet exemplářů .....

2. Autor prohlašuje, že vytvořil samostatnou vlastní tvůrčí činností dílo shora popsané a specifikované. Autor dále prohlašuje, že při zpracovávání díla se sám nedostal do rozporu s autorským zákonem a předpisy souvisejícími a že je dílo dílem původním.
3. Dílo je chráněno jako dílo dle autorského zákona v platném znění.
4. Autor potvrzuje, že listinná a elektronická verze díla je identická.

Článek 2

Udělení licenčního oprávnění

1. Autor touto smlouvou poskytuje nabyvateli oprávnění (licenci) k výkonu práva uvedené dílo nevýdělečně užít, archivovat a zpřístupnit ke studijním, výukovým a výzkumným účelům včetně pořizovaní výpisů, opisů a rozmnoženin.
2. Licence je poskytována celosvětově, pro celou dobu trvání autorských a majetkových práv k dílu.
3. Autor souhlasí se zveřejněním díla v databázi přístupné v mezinárodní síti
   - ☑ ihned po uzavření této smlouvy
   - ☐ 1 rok po uzavření této smlouvy
   - ☐ 3 roky po uzavření této smlouvy
   - ☐ 5 let po uzavření této smlouvy
   - ☐ 10 let po uzavření této smlouvy

   (z důvodu utajení v něm obsažených informací)
4. Nevýdělečné zveřejňování díla nabyvatelem v souladu s ustanovením § 47b zákona č. 111/1998 Sb., v platném znění, nevyžaduje licenci a nabyvatel je k němu povinen a oprávněn ze zákona.

Článek 3

Závěrečná ustanovení

1. Smlouva je sepsána ve třech vyhotoveních s platností originálu, přičemž po jednom vyhotovení obdrží autor a nabyvatel, další vyhotovení je vloženo do VŠKP.
2. Vztahy mezi smluvními stranami vzniklé a neupravené touto smlouvou se řídí autorským zákonem, občanským zákoníkem, vysokoškolským zákonem, zákonem o archivnictví, v platném znění a popř. dalšími právními předpisy.
3. Licenční smlouva byla uzavřena na základě svobodné a pravé vůle smluvních stran, s plným porozuměním jejímu textu i důsledkům, nikoliv v tísni a za nápadně nevýhodných podmínek.
4. Licenční smlouva nabývá platnosti a účinnosti dnem jejího podpisu oběma smluvními stranami.

V Brně dne: .....

....

Nabyvatel Autor

Abstrakt:

Tato diplomová práce se zabývá problematikou ovládání mikroposuvu. Jejím cílem je jednak navrhnout zařízení, které bude s nastavitelnou rychlostí posuvu a velikostí kroku ovládat 3D piezoposuv. A dále vytvořit program pro osobní počítač, který bude přenášet do zařízení uživatelem definovaný profil posuvu. Praktická část práce popisuje konkrétní realizaci zařízení založenou na mikrokontroleru Microchip 18F4550.

Abstract:

This thesis deals with control of a micropositioning device. Aim of this study was to design a device that will control the piezzo-shift in three dimensions with adjustable speed and step-size. A further goal was to create a software for which allow to employ a user-defined shift profile to control the device. Actual construction of the device, based on a microcontroler Microchip 18F4550, is described in the practical section of the work.

Klíčová slova:

Ovládání, mikroposuv, piezo.

Keywords:

Control, micropositioning, piezo.

Bibliografická citace díla:

CHMELÍK, J. Zařízení pro ovládání mikroposuvu. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, 2008. 62 s. Vedoucí diplomové práce Ing. Jaroslav Kadlec, Ph.D.

Prohlášení autora o původnosti díla:

Prohlašuji, že jsem tuto vysokoškolskou kvalifikační práci vypracoval samostatně pod vedením vedoucího diplomové práce, s použitím odborné literatury a dalších informačních zdrojů, které jsou všechny citovány v práci a uvedeny v seznamu literatury. Jako autor uvedené diplomové práce dále prohlašuji, že v souvislosti s vytvořením této diplomové práce jsem neporušil autorská práva třetích osob, zejména jsem nezasáhl nedovoleným způsobem do cizích autorských práv osobnostních a jsem si plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestněprávních důsledků vyplývajících z ustanovení § 152 trestního zákona č. 140/1961 Sb.

V Brně dne 26. 5. 2008

…….

Poděkování:

Chtěl bych poděkovat vedoucímu mé práce Ing. Jaroslavu Kadlecovi, Ph.D. za odbornou pomoc a rady při řešení této diplomové práce.

OBSAH

1 Úvod

Cílem mé diplomové práce je sestavit zařízení pro ovládání piezo posuvu. Tento druh posuvu využívá ke své funkci opačný piezoelektrický jev, tj. schopnost krystalu měnit svůj tvar a rozměry v závislosti na přivedeném napětí. Výhodou takových posuvů je schopnost měnit své rozměry o velmi malé vzdálenosti (až desítky pm). Takovéto posuvy mají oblast použití např. v metrologii, skenovací mikroskopii, nanopolohování nebo skenovací interferometrii.

Princip piezo posuvu spočívá v přivedení napětí o velikosti řádově stovek voltů a definovaného průběhu na piezoelektrický materiál. Vlivem působení napětí dochází k mechanické deformaci tohoto materiálu. Deformace je pomocí vhodného mechanického uspořádání převedena na mechanický pohyb. Cílem této práce je tedy sestrojit zařízení, které bude generovat toto napětí.

Práci je rozdělena na tři hlavní části. V první části je přiblížena problematika piezoelektrického jevu a popis piezo posuvu. Dále je v této části uvedena stručná charakteristika použité USB sběrnice.

Druhá část práce se skládá z několika bloků popisujících zdrojovou, analogovou a digitální část navrženého zařízení. Dále je zde proveden rozbor programu pro mikrokontroler a popis ovládací aplikace pro osobní počítač. Je zde také nastíněno mechanické provedení zařízeni.

V poslední části práce jsou uvedeny naměřené hodnoty, které v závěru porovnávám s hodnotami teoretickými. Je zde uvedeno do jaké míry se podařilo splnit cíle tohoto projektu a zamýšlím se také nad dalším možným vývojem zařízení.

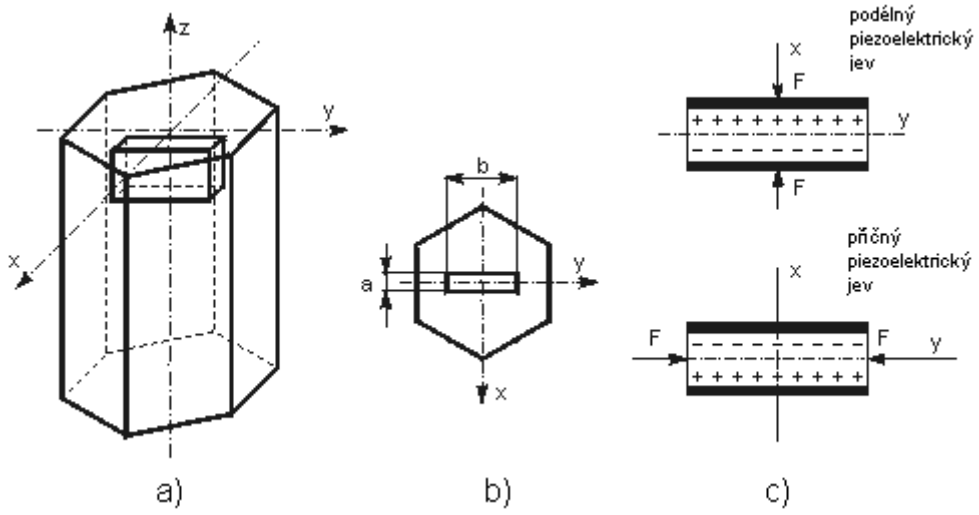
2 Teoretický rozbor

2.1 Piezoelektrický jev

Piezoelektrický efekt objevili roku 1880 bratři Pierre a Jacques Curieové, kteří zjistili, že v některých anizotropních krystalech se při mechanickém namáhání generují orientované dipóly a vzniká elektrické napětí. Tento jev byl nazván přímý piezoelektrický jev, podle řeckého slova piedzó (tlačit). O rok později objevili opačný piezoelektrický jev, u něhož vnější elektrické pole vyvolává deformaci krystalu.

Dielektrická polarizace některých látek je tedy vázána na jejich elastickou deformaci – tlak, tah, ohyb. Vzniklý elektrický náboj je přímo úměrný působící síle a konstanta úměrnosti je tzv. piezoelektrická konstanta. Napětí se odebírá z elektrod vyvedených přímo z krystalu.

Piezoelektrický element získáme z krystalu křemene (obr. 1) tak, že vyřízneme destičku, jejíž hrany budou rovnoběžné s jednotlivými osami krystalu (x – osa elektrická, y – osa mechanická, z – osa optická).



Obr. 1: a) Krystal křemene, b) Výbrus elementu, c) Piezoelektrický jev [1]

Působí-li na křemennou destičku rovnoměrně rozložená síla F_x ve směru elektrické osy x, hovoříme o tzv. podélném piezoelektrickém jevu. Velikost náboje Q, vznikajícího na stěnách kolmých k elektrické ose, bude

$$Q = K_p \cdot F_x , \qquad (1)$$

kde K_p je piezoelektrická konstanta. Z rovnice (1) je vidět, že velikost náboje vzniklého při působení sil podél elektrické osy x nezávisí na rozměrech krystalového výbrusu.

Působí-li na krystal síla F_y ve směru mechanické osy y, vznikají náboje opět na plochách kolmých na elektrickou osu, avšak směr polarizačního vektoru je záporný a velikost náboje závisí na geometrických rozměrech krystalu. Hovoříme o tzv. příčném piezoelektrickém jevu. Velikost náboje Q je dána vztahem

$$Q = -K_p \cdot F_y \cdot \frac{b}{a}, \tag{2}$$

kde a, b jsou rozměry destičky.

Průmyslově využitelnými látkami vykazujícími piezoelektrické vlastnosti jsou například fosforečnan amonný, titaničitan barnatý, Seignettova sůl. Piezoelektriky jsou i některé přírodní krystaly – již zmíněný křemen a turmalín. Jako technická piezoelektrika se využívají keramické materiály na bázi tuhých roztoků oxidů olova (Pb), zirkonu (Zr) a titanu (Ti) – tzv. PZT keramika. Tyto látky se pro jejich piezoelektrické vlastnosti průmyslově využívají přibližné od poloviny 20. století, kdy se započalo s jejich intenzivním výzkumem. Používají se například jako indikátory tlaku, snímače vibrací, piezotransformátory, frekvenční keramické filtry nebo elektroakustické měniče.

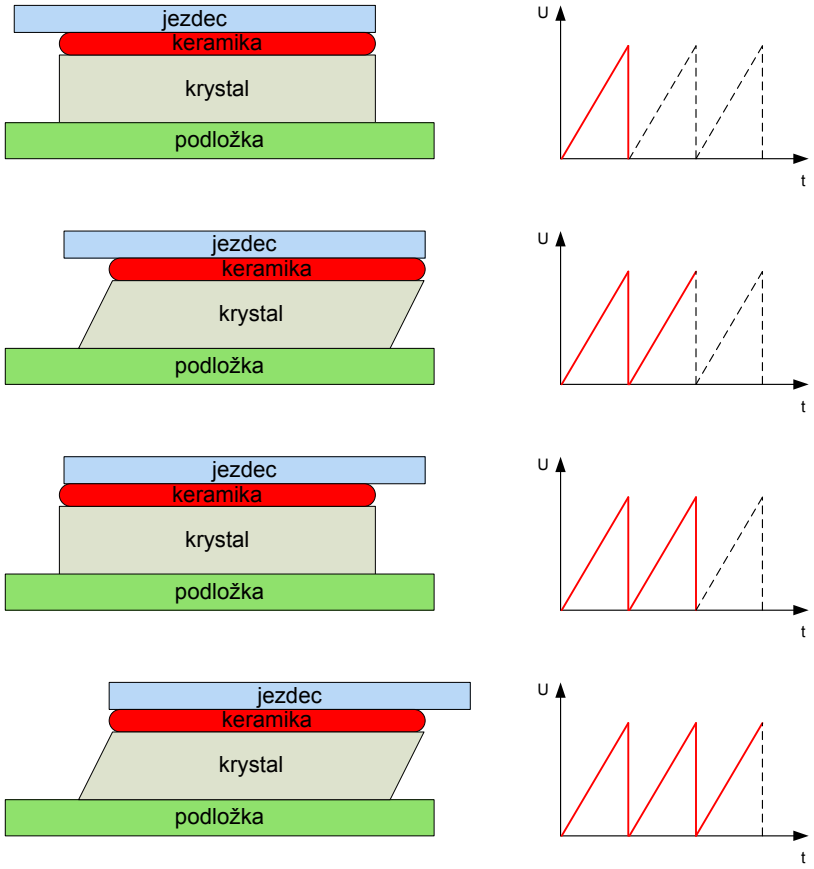
Piezoelektrický jev s postupujícím technickým zdokonalováním piezotechnologií zasahuje kromě elektrotechniky i do řady dalších oblastí: optiky (jemné posuny, skenovaní mikroskopie), automobilní a spotřební elektrotechniky (zapalování zážehových spalovacích motorů, domácí spotřebiče) a také medicíny (zdroje ultrazvuku pro lékařské diagnostické přístroje).

2.2 Piezoposuv

Základní fyzikální princip, na kterém jsou postaveny piezoelektrické motory, je obrácený piezoelektrický jev, tj. přivedením elektrického napětí (řádově ve stovkách voltů) na některé krystaly či keramické materiály lze dosáhnout jejich mechanické deformace. Taková deformace však obvykle činí pouhé zlomky celkového rozměru materiálu, typicky 0,1 %. Pro použití v praxi je tedy důležité buď deformaci piezoelektrického materiálu mechanicky zesílit, a nebo výsledný pohyb manipulátoru složit z několika dílčích kroků poskytnutých piezoelektrickým materiálem.

Na obrázku 2 je ukázán princip piezoelektrických motorů typu stick-slip (přilepit-proklouznout). Zajímavostí tohoto motoru je, že využívá tření jako hlavní a žádaný mechanismus. S lineárně rostoucím napětím přiloženým na piezoelektrický materiál polarizovaný do střihu se materiál deformuje a volně položený jezdec se pohybuje doprava. Napětí pomalu roste až do velikosti několika set voltů, načež velmi strmě poklesne a jezdec se

tak pro svou setrvačnost nestačí vrátit a zůstane v nové pozici. Opakováním tohoto cyklu je možné dosáhnout vychýlení téměř libovolného rozsahu. Délka jednoho kroku bývá řádově jednotky nm. Popisovaný motor lze tedy chápat jako krokový motor s uvedenou délkou kroku.



Obr. 2: Piezoposuv typu stisk-slip

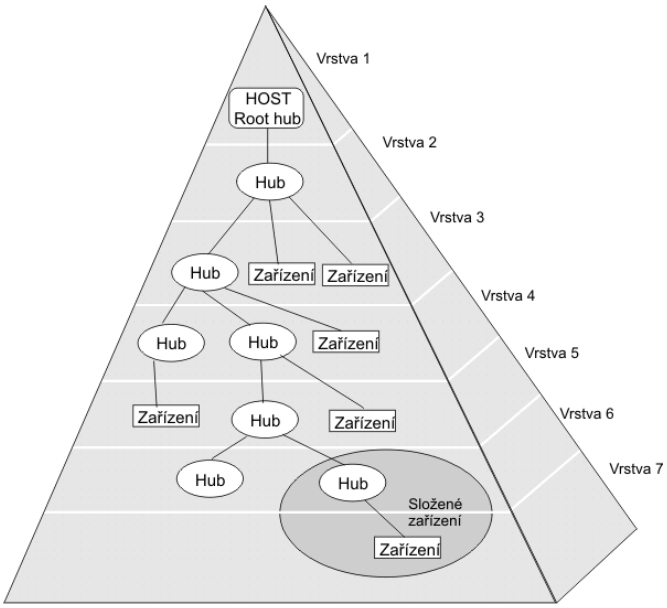
2.3 USB

USB (Universal Serial Bus) je moderní sériová sběrnice, která byla standardizována v roce 1995, resp. v revizi 2.0 v roce 2000, firmami Compaq, Hewlett-Packard, Intel, Lunect, Microsoft, Nec a Philips. Hlavním důvodem vzniků této sběrnice bylo sjednocení a nahrazení tehdy používaných způsobů připojení periferií k počítači (sériový a paralelní port, PS/2, GamePort apod.). Byly definovány cíle, které musí sběrnice splňovat:

- Jednoduché připojování periférií
- Levné řešení podporující přenos do 480Mbps (v rev. 2.0)
- Plná podpora pro datové přenosy v reálném čase (hlas, zvuk, komprimované video)
- Podpora isochroních i asynchronních přenosů
- Poskytnout standardní rozhranní, které bude schopné se rychle rozšiřovat do ostatních zařízení
- Rozšířit možnosti PC a umožnit vznik nových zařízení

2.3.1 Topologie

USB dle specifikace [2] využívá vrstevnou hvězdicovou topologii, kde je v centru každé hvězdice tzv. USB hub. K tomuto hubu může být připojen buď další hub (na další úrovni) nebo koncové zařízení. Topologie USB je vyobrazena na následujícím obrázku.



Obr. 3: Architektura USB [2]

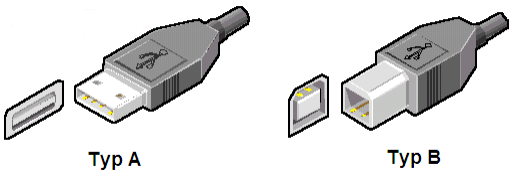
USB sběrnice obsahuje jeden tzv. kořenový rozbočovač (Root hub), který je považován za nejvyšší (první) úroveň, a k němuž jsou připojeny další huby a zařízení. Rozhraní mezi USB systémem a hostitelským počítačem je nazýváno hostitelský řadič. Tento řadič může být implementován hardwarově nebo softwarově. Kořenový rozbočovač je integrován spolu s hostitelským řadičem do hostitelského systému.

S ohledem na zpoždění signálu v kabelech a hubech povoluje specifikace maximálně sedm úrovní včetně kořenové vrstvy. To znamená, že mezi kořenovým rozbočovačem a koncovým zařízením může být zapojeno maximálně pět rozbočovačů.

2.3.2 Elektrické parametry

USB sběrnice využívá čtyři vodiče. Dva z nich slouží jako napájení, po dalších dvou jsou přenášena diferenciálně vlastní data. Díky tomu má USB sběrnice i při vysokých přenosových rychlostech značnou odolnost proti šumu a proti rušení. USB 2.0 specifikuje parametry kabelů pro propojování zařízení. Pro full / high speed je vyžadován stíněný krouceny kabel maximální délky 5 metrů, pro low speed není stínění vyžadováno a délka kabelu je omezena na tři metry.

Konektory se na straně hostitele a na straně zařízení liší. Nelze tedy spojit dva hostitele nebo dvě zařízení k sobě. Typ A je na straně hostitele a typ B je na straně zařízení. U miniaturních zařízení se nemusí používat propojovací A-B kabel, ale zařízení je přímo opatřeno konektorem typu A (samec).



Obr. 4: USB konektory

2.3.3 Definice rychlosti

USB sběrnice používá tři rychlosti přenosu dat. Vysvětlení jednotlivých rychlostí a oblasti použití jsou shrnuty v následující tabulce.

Tabulka 1: Rychlosti USB

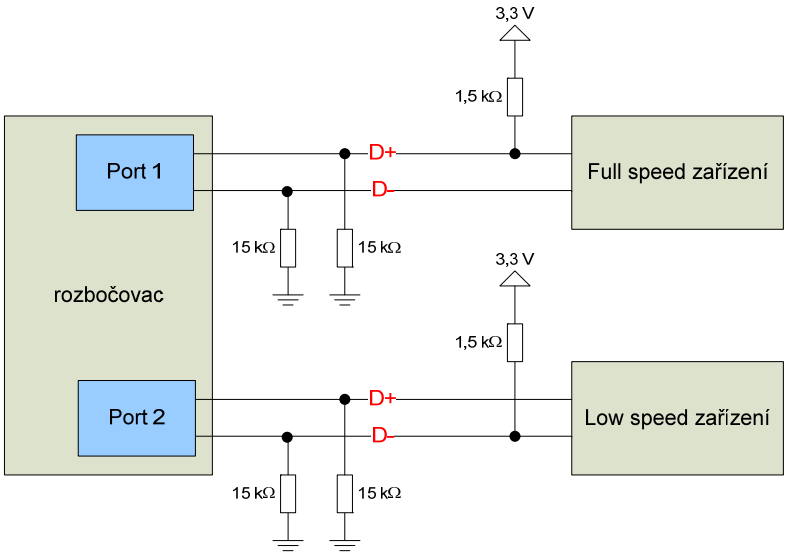
Režim	Rychlost	Oblast použití
Low speed	1,5 Mb/s	klávesnice, myš, herní ovladače
Full speed	12 Mb/s	mikrofony, reproduktory, komprimované video
High speed	480 Mb/s	pevné disky, nekomprimované video

Uvedené přenosové rychlosti platí pro jedno zařízení. Je-li k počítači (root hubu) připojeno více zařízení, rozděluje se šířka pásma mezi jednotlivá zařízení. Výhodou USB sběrnici je, že v každém segmentu komunikuje na nejvyšší možné rychlosti jakou hub, resp. připojené zařízeni podporují. Např. je-li ke kořenovému high speed hubu zapojen další high speed hub a k němu pomalé zařízení, tak komunikace mezi high speed zařízeními probíhá v režimu high speed.

Použitou přenosovou rychlost definují sama zařízení:

- Low speed zařízení připojuje pull-up rezistor 1,5 kΩ mezi D- a 3,3 V
- Full speed zařízení připojuje pull-up rezistor 1,5 kΩ mezi D+ a 3,3 V
- High Speed zařízení se detekují stejně jako zařízení Full speed s tím, že změna rychlosti se řeší programově.

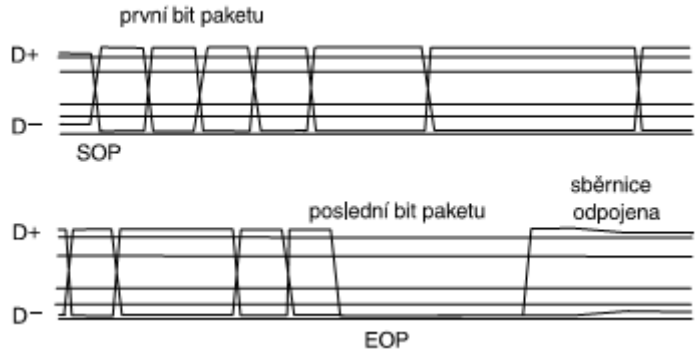
Připojení pull-up rezistorů na D+ nebo D- zároveň hubu sděluje, že je připojeno zařízení, jinak jsou linky taženy k zemi pomocí 15 kΩ pull-down rezistorů (viz obr. 5).



Obr. 5: Připojení USB zařízení k rozbočovači

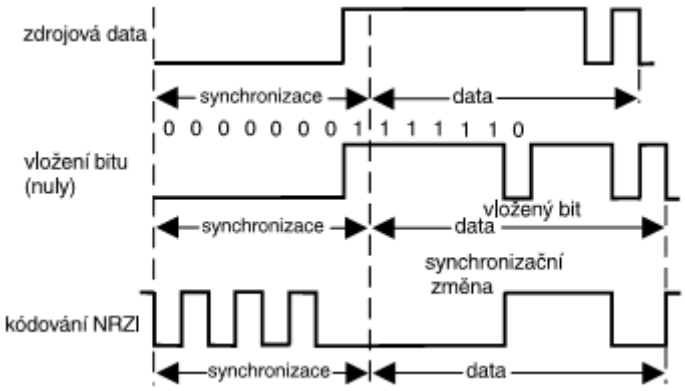
2.3.4 Sériový přenos

Datový signál je generován budičem s diferenčním výstupem, takže signál ve vodiči D+ je inverzní signálu ve vodiči D- a naopak. Data jsou přenášena v paketech (obr. 6). Začátek paketu je označen SOP (Start Of Packet) a konec paketu EOP (End Of Packet).



Obr. 6: Průběh napětí při přenosu dat

Data jsou v systému USB kódována metodou NRZI (Non Return to Zero Invert). Logická nula je tudíž reprezentována změnou napěťových úrovní na datových vodičích a logická jednička absencí této změny. Protože paralelně s daty není přenášen žádný taktovací signál, je na začátku každého paketu vyslán tzv. synchronizační vzorek, který má za úkol synchronizovat taktovací generátor přijímače. Synchronizační vzorek je tvořen sedmi nulami následovanými jedničkou v kódování NRZI. Pro udržení synchronizace se vždy, následuje-li po sobě šest jedniček, vloží nula – vložený bit (bit stuffing). Tímto způsobem je zajišťována potřebná změna signálu. Na obr. 7 je vzorek přenášených dat a jeho zakódování metodou NRZI s využitím bit stuffingu.

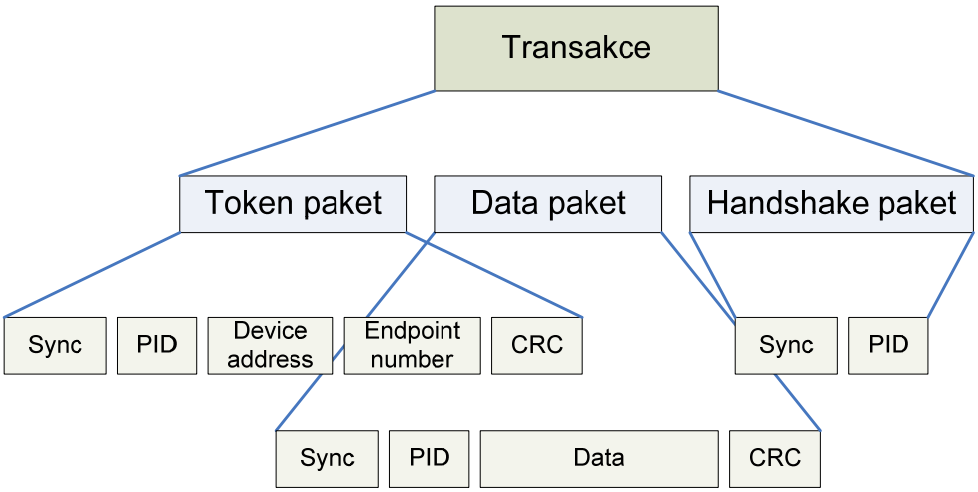


Obr. 7: Synchronizace při přenosu dat

2.3.5 Transakce

USB využívá pro přenos dat polling, tj. systém, ve kterém jsou jednotlivá připojená zařízení vyzývána k vysílání. Většina sběrnicových transakcí (přenosů dat) sestává z vyslání tří paketů (obr. 8). Každá transakce začíná tím, že Host Controller vyšle USB paket popisující typ a směr přenosu, adresu zařízení a číslo koncového bodu (endpoint) v zařízení. Tento paket je označen jako token paket. USB zařízení, které rozpozná svou adresu se připraví k přenosu. Směr přenosu, tedy zda jde o přenos dat ze zařízení do hostitelského systému nebo z

hostitelského systému do zařízení, je určen token paketem. Poté zdroj dat (zařízení nebo systém) vyšle datový paket, nebo oznámí, že nemá žádná data k vyslání. Datové pakety jsou zabezpečeny 16bitovým CRC kódem a ostatní pakety jsou zabezpečeny 5bitovým CRC kódem. Transakce bývá ukončena tím, že příjemce (cíl dat) vyšle handshake paket, kterým potvrdí úspěšnost přenosu. Některé transakce mezi hostitelským systémem a hubem se skládají ze čtyř paketů. Transakce tohoto typu jsou používány pro řízení datových přenosů mezi hostitelským systémem a full / low speed zařízeními.



Obr. 8: Význam paketů při přenosu dat

2.3.6 Roura

Pomyslná cesta pro datové přenosy mezi hostitelským zařízením a koncovým bodem v zařízení je nazývána rourou (pipe). Existují dva typy rour:

- pro datové proudy (streams)
- pro zprávy (messages)

Narozdíl od zpráv nemá datový proud pevně definovanou strukturu. Roura má dále přiřazené některé parametry jako jsou šířka přenosového pásma, typ přenosu a charakteristiky koncového bodu. Většina rour je vytvořena v okamžiku konfigurace USB zařízení. Jedna roura pro přenos zpráv, tzv. Default Control Pipe, existuje ihned po připojení zařízení a poskytuje přístup ke konfiguračním, stavovým a řídícím informacím zařízení.

USB umožňuje výměnu dat a řídících informací mezi hostitelským systémem a koncovým zařízením pomocí množiny jednosměrných či obousměrných rour. Každá z těchto rour je připojena k jednomu koncovému bodu daného zařízení. Komunikace v jednotlivých rourách je na sobě navzájem nezávislá. Pro USB jsou definovány čtyři typy přenosu dat:

- Řídicí přenosy (Control transfer) - používají se k řízení hardware, pracují s vysokou prioritou a automatickým zabezpečením chyb. Přenosová rychlost je vysoká, na jeden dotaz lze přenést až 64 bajtů.
- Přenos přes přerušení (Interrupt-Transfer) - používají zařízení, která periodicky vysílají menší množství dat (například myš nebo klávesnice). Počítač se periodicky dotazuje na nová data. Typicky se najeden dotaz přenese 8 bajtů.
- Hromadný přenos (Bulk-Transfer) - je vhodný pro přenos velkých množství dat se zabezpečením. Priorita přenosu je nízká, takže tento typ není vhodný pro časově kritické operace (typické použití: tiskárna, skener, externí ZIP).
- Izochronní přenos (Isochronous-Transfer) - je vhodný pro přenos velkých množství dat definovanou rychlostí (nejvyšší priorita) bez jejich zabezpečení. Je vhodný pro systémy, kde větší problém představuje výpadek přenosu než chyba v přenosu (například externí zvukové karty, web kamery)

2.3.7 Koncový bod

Koncový bod (endpoint) je část USB zařízení, která stojí na jednom konci komunikačního toku mezi hostitelem a zařízením. Každé logické zařízení se skládá z několika nezávislých koncových bodů. Tomuto logickému zařízení je během připojování přiřazena jednoznačná identifikace v rámci sběrnice. Každý koncový bod v rámci zařízení má výrobcem přiřazený kód, nazývaný číslo koncového bodu (endpoint number) v rozsahu 0-15. Endpointy mají zároveň návrhem pevně daný směr komunikace (příjem / vysílání dat). Kombinace identifikátoru zařízení, čísla endpointu a směru komunikace dává dohromady jednoznačné určení endpointu v rámci USB sběrnice. V jednom zařízení se tedy mohou vyskytnout dva endpointy se stejným číslem a různým směrem přenosu dat. Koncový bod je popsán:

- Číslem koncového bodu
- Typem přenosu
- Směrem přenosu
- Požadavky na chování obsluhy chyb
- Požadavky na frekvenci a latenci přístupů ke sběrnici
- Požadavkem na šířku přenosového pásma
- Maximální velikostí paketu, kterou je schopen endpoint přijmout nebo odeslat

Koncový bod 0 (endpoint zero) musí být implementován v každém USB zařízení a slouží k prvotní konfiguraci zařízení. USB systém používá endpoint 0 k inicializaci a k některým generickým úkonům, jako jsou konfigurace logického zařízení. Zároveň tento endpoint poskytuje informace o konfiguraci a stavu zařízení. Endpoint 0 podporuje řídící přenosy. Ostatní koncové body mohou být implementovány USB zařízením podle potřeby. Tyto koncové body jsou po připojení v nedefinovaném stavu a nesmí k nim být přistupováno až do doby, než je zařízení nakonfigurováno.

2.3.8 Enumerace

Každé USB zařízení se po svém připojení přihlašuje do operačního systému počítače. Při tom poskytuje svůj popis ve formě deskriptoru, který nese podrobné informace o zařízeni (sériové číslo, požadavek na proudový odběr, popis zařízení, atd.). Tyto informace jsou důležité pro operační systém, který na jejich základě hledá vhodný ovladač. Tento proces získávání prvotních informací se nazývá enumerace a jeho průběh je následující:

1. Zařízení je detekováno na sběrnici – zařízení je ve stavu POWERED.
2. Zařízení je resetováno a přechází tak do stavu DEFAULT. Nyní zařízení přijímá a odpovídá na příkazy na výchozím řídicím endpointu 0 na adrese 0.
3. Host kontrolér vyšle příkaz pro nastavení adresy zařízení, které se tak dostává do stavu ADDRESSD.
4. Hostitelský software nyní přečte konfigurační deskriptory, zvolí vhodnou konfiguraci a zařízení se dostává do stavu CONFIGURED. Nyní muže započít komunikace na všech endpointech.

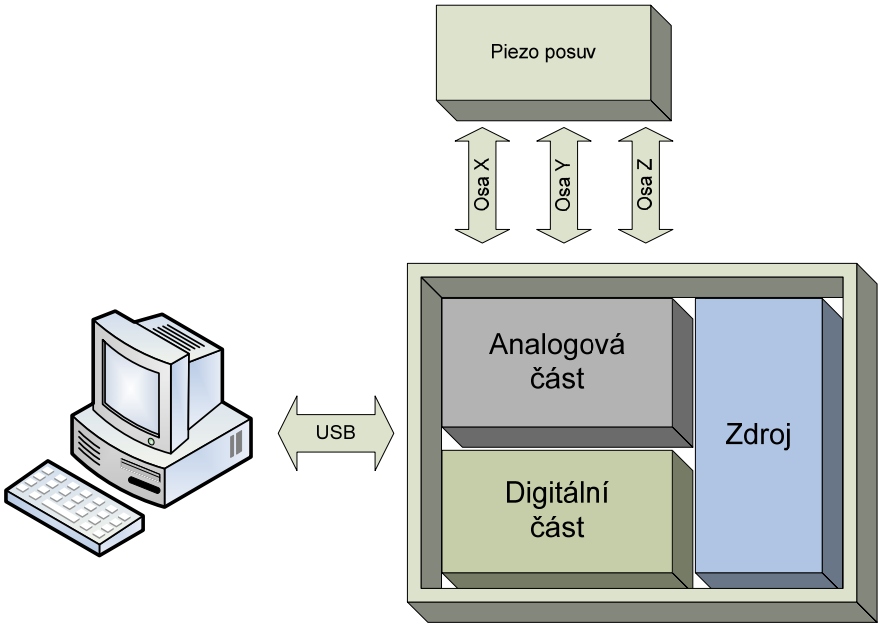
2.3.9 Deskriptor

Deskriptor je datová struktura s definovaným formátem, pomocí které USB zařízení informuje o svých parametrech nadřazený software. Norma specifikuje 5 základních typů deskriptorů:

- deskriptory zařízení (Device Descriptors)
- konfigurační deskriptory (Configuration Descriptors)
- deskriptory rozhraní (Interface Descriptors)
- Endpoint deskriptory
- String deskriptory

3 Řešená problematika

Způsob ovládání mikroposuvu vychází z jeho piezoelektrické podstaty, tj. přivedením napětí o určité amplitudě a frekvenci. Mnou navržený přístroj je schopen nezávisle ovládat 3 osy. Pro každou osu je možné nezávislé nastavení amplitudy (do 240V), délky jednoho impulsu a kmitočtu generovaného signálu (1 Hz – 500 Hz po 1Hz). Dále je možné nastavit počet jednotlivých impulsů. Na obr. 9 je blokové schéma přístroje. Na následujících stranách budou popsány jeho jednotlivé části.



Obr. 9: Blokové schéma přístroje

3.1 Zdroj

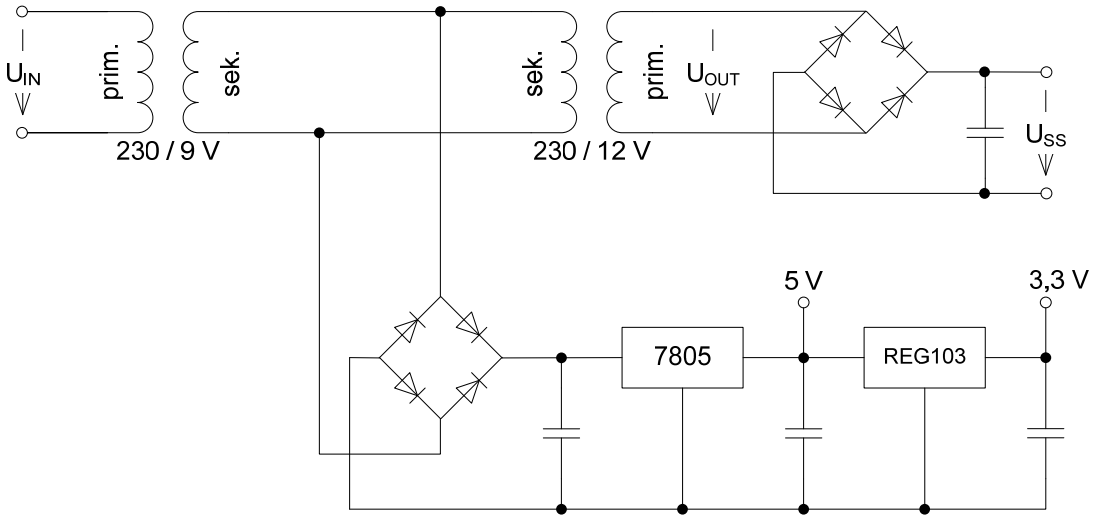
Protože pro piezo posuv bylo nutné zajistit galvanické oddělení od sítě, je v zapojení využito kaskádního zapojení dvou transformátorů (obr. 10). Pokud na vstup takto zapojeného zdroje přivedeme napětí 230 V, pak hodnota výstupního napětí bude:

$$U_{OUT} = \frac{9 \cdot U_{IN}}{12} = \frac{9 \cdot 230}{12} = \underline{\underline{172{,}5V}} \tag{3}$$

Za dvojící transformátorů následuje usměrňovač. Hodnotu stejnosměrného napětí za usměrňovačem a filtračním kondenzátorem vypočítáme dle (4). Tato hodnota tedy bude maximální možná hodnota napájecího napětí pro piezo posuv.

$$U_{SS} = U_{OUT} \cdot \sqrt{2} = 172{,}5 \cdot \sqrt{2} = \underline{\underline{244V}} \tag{4}$$

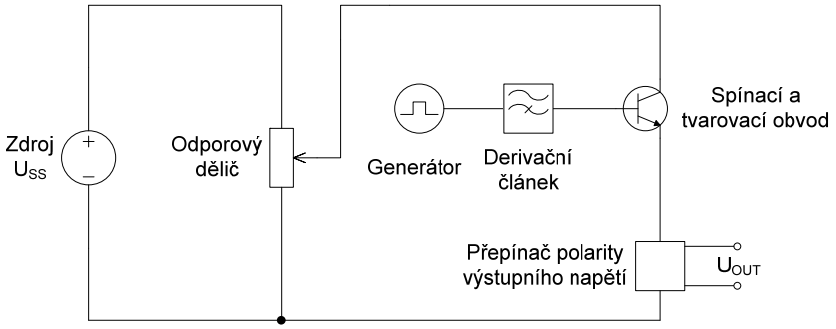
Pro napájení elektrických obvodů je dále nutné vytvořit napětí 3,3 V a 5 V. Tyto napětí jsem vytvořil postupným zapojením dvou stabilizátoru. Vstupem stabilizátoru 7805 [3] je usměrněné napětí odebírané ze sekundárního vinutí prvního transformátoru. Výstupem je napětí 5 V, které se využívá pro napájení relé a jako vstupní napětí stabilizátoru REG103 [4]. Nutnost dvou napájecích napětí vychází z maximální hodnoty napájecího napětí displeje (3,6 V) a napájecího napětí pro relé (5 V).



Obr. 10: Schéma zapojení zdroje

3.2 Analogová část

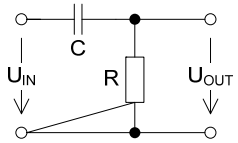
Jelikož zařízení má být schopno nezávisle ovládat tři osy mikroposuvu, tak se analogová část skládá ze tří identických bloků. Zapojení takového bloku je na obr. 11. Usměrněné napětí ze zdroje je přivedeno na odporový dělič, na kterém nastavíme požadovanou hodnotu výstupního napětí. Toto napětí je dále tvarováno a spínáno s nastavitelnou frekvencí.



Obr. 11: Blokové schéma analogové části

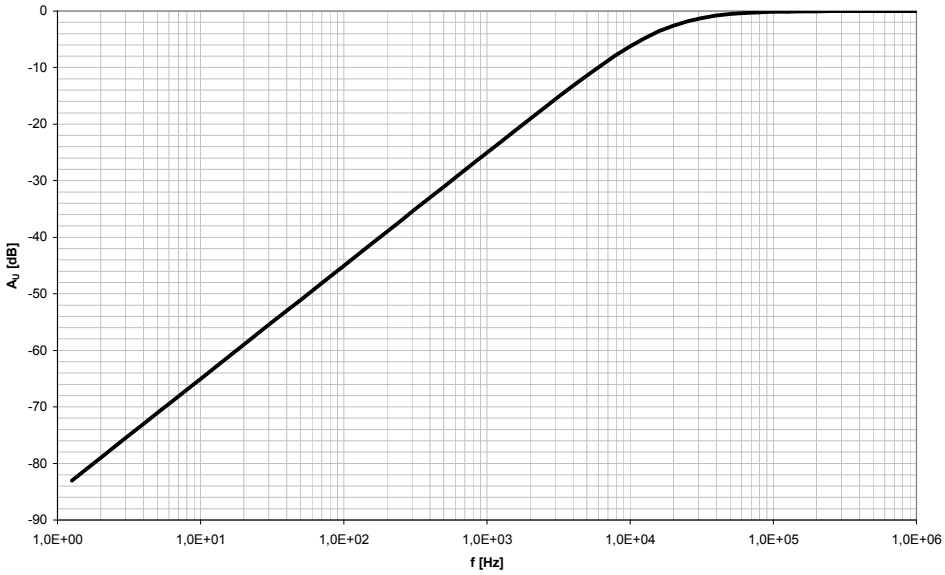
3.2.1 Derivační článek

Derivační článek je obvod, který provádí matematickou funkci derivování – výstupní napětí je derivací vstupního napětí. Článek obsahuje nejméně jednu kmitočtově závislou součástku (kondenzátor, cívka). Nejjednodušším zapojením je pasivní zapojení využívající jeden kondenzátor a jeden rezistor (obr. 12).



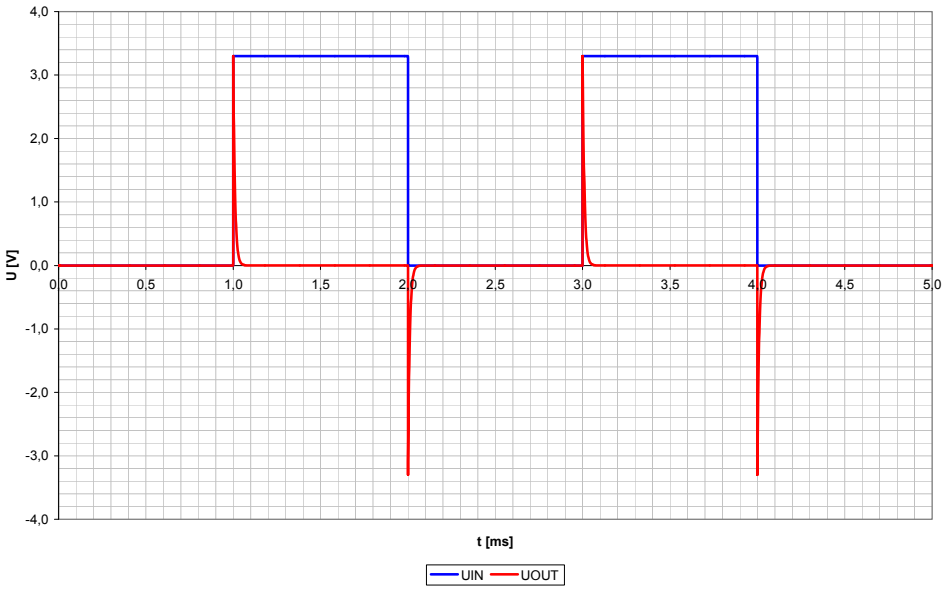
Obr. 12: Derivační článek

Derivační článek má frekvenční charakteristiku horní propusti. Kmitočtová závislost pro mnou použité hodnoty *C=1 nF*, a *R=10 kΩ* je na následujícím obrázku.



Obr. 13: Kmitočtová charakteristika derivačního článku

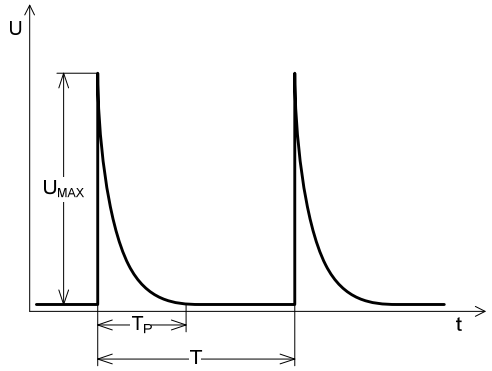
Derivační článek je buzen signálem obdélníkového průběhu generovaným pomocí mikrokontroleru. Na obrázku 14 je průběh vstupního a výstupního napětí. Za povšimnutí stojí záporný impulz výstupního napětí při sestupné hraně vstupního signálu. Tento fakt je nutné brát v úvahu při výběru spínacího tranzistoru. Jeho maximální povolené napětí emitor – báze musí být větší, než hodnota záporného impulsu, která je v našem případě 3,3 V. Použitý tranzistor 2N3439 tuto podmínku splňuje, protože jeho U_{EBmax}=7 V.



Obr. 14: Průběhy vstupního a výstupního napětí derivačního článku

3.2.2 Tvarovací a spínací obvod

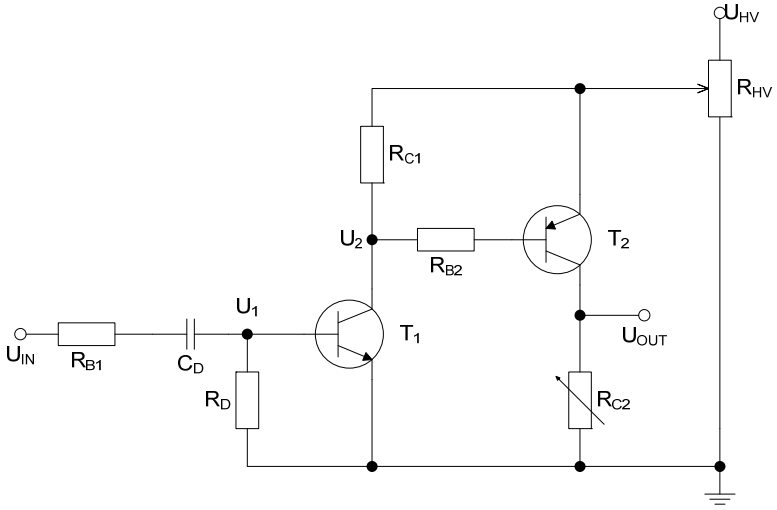
Požadovaný průběh signálu pro ovládání mikroposuvu je na následujícím obrázku. Perioda T, resp. frekvence takového signálu je nastavitelná od *1 Hz* do *500 Hz* s krokem *1 Hz*. Maximální hodnota U_{MAX} je nastavitelná pomocí odporového děliče a lze jí měnit v rozmezí *120 V – 244V*. Délku impulsu T_P můžeme měnit v řádech stovek mikrosekund.



Obr. 15: Požadovaný průběh signálu ovládající mikroposuv

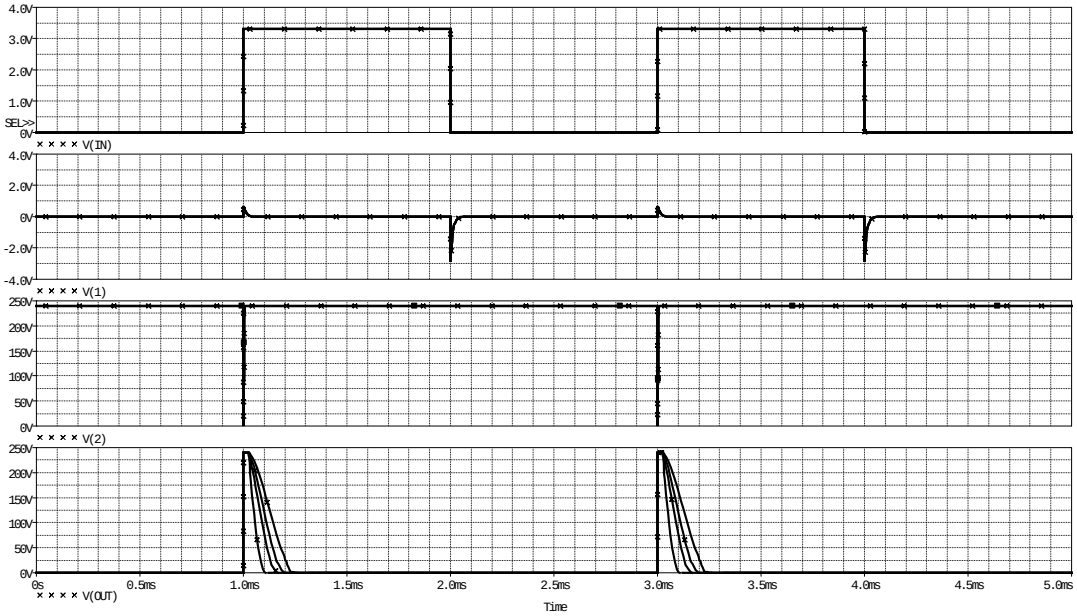
Schéma zapojení tvarovacího a spínacího obvodu je na obrázku 16. Skládá se z derivačního článku, dvojce tranzistorů 2N3439 [5] a 2N5416 [6] sloužících jako spínač a několika rezistorů. Rezistory R_{B1}, R_{B2}, R_{C1} nastavují pracovní body tranzistorů. Potenciometrem R_{HV} nastavujeme maximální hodnotu výstupního napětí a rezistorem R_{C2} nastavujeme délku impulsu. Čím větší bude hodnota tohoto rezistoru, tím déle se bude vybíjet

kapacita tranzistoru T_2, a tím delší bude impuls. Závislost délky impulsu na velikosti rezistoru R_{C2} je na obrázku 18.

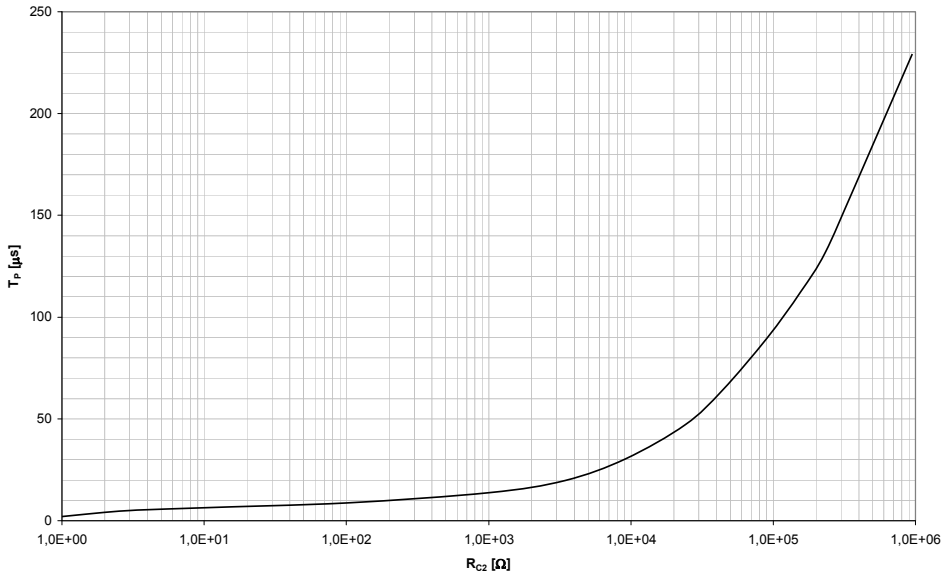


Obr. 16: Zapojení analogové části

Obdélníkový signál generovaný mikrokontrolerem (obr. 17 – první průběh) prochází derivačním článkem a je přiveden na bázi tranzistoru T_1 (obr. 17 – druhý průběh). Se vzrůstajícím napětím na bázi se tranzistor bude více otevírat. Čím více bude tranzistor otevřený, tím bude menší napětí na jeho kolektoru (obr. 17 – třetí průběh), a tím více se bude otevírat tranzistor T_2. Pokud se otevře tranzistor T_2, bude jím procházet proud, který vytvoří úbytek napětí na odporu R_{C2}. Výstupní napětí je potom rovno tomuto úbytku (obr. 17 – čtvrtý průběh). Simulace proběhla pro různé hodnoty odporu R_{C2}.



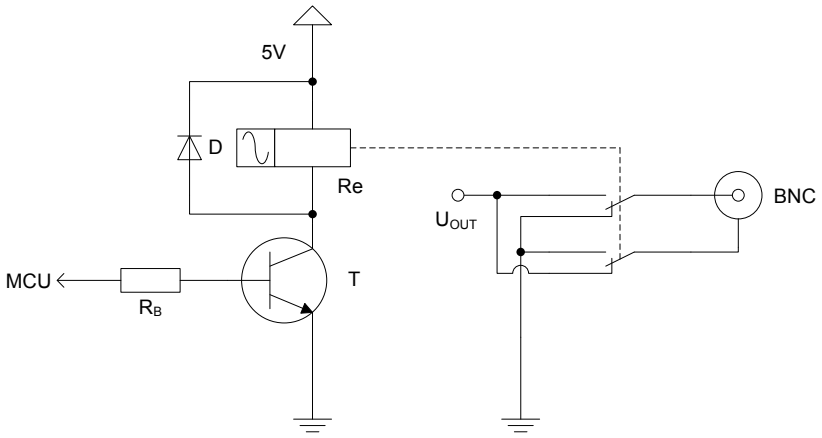
Obr. 17: Průběhy napětí v analogové části



Obr. 18: Závislost délky impulsu na velikosti rezistoru RC2

3.2.3 Přepínač polarity výstupního napětí

Polarita výstupního napětí (resp. směr posuvu) je přepínána pomocí relé, které je zapojeno tak, že v jednom stavu připojí výstupní napětí na střední vodič a zem na plášť konektoru a ve druhém stavu naopak.



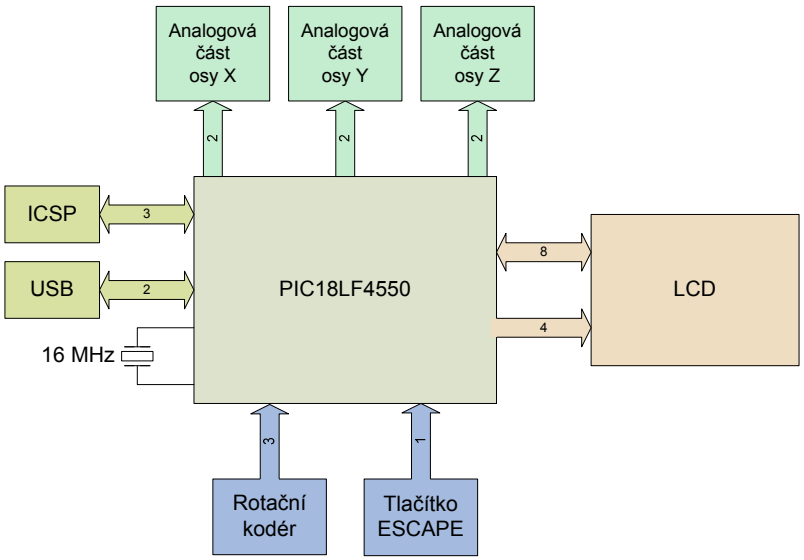
Obr. 19: Schéma zapojení přepínače směru posuvu

3.3 Digitální část

Hlavním obvodem digitální části je PIC18LF4550 [7]. Jedná se o 8bitový mikrokontroler firmy Microchip. Mezi hlavní důvody proč jsem vybral právě tento obvod patří:

1. možnost použití nízkého napájecího napětí ($V_{DD\min} = 2V$)
2. 3x 16bitový časovač
3. integrované USB
4. 34 I/O portů
5. 32KB FLASH paměť

Na obrázku 20 je vyobrazeno připojení jednotlivých bloků k mikrokontroleru. Pomocí šipek a v nich umístěných čísel je znázorněn směr a počet datových vodičů spojujících jednotlivé bloky s mikrokontrolerem. Pro zobrazení aktuálního stavu zařízení, tj. frekvence generovaných kmitočtů, směr posuvu, počet pulsů atd., je k mikrokontroleru připojen grafický display W128-6X9. Aby bylo možné měnit nastavení mikroposuvu jsou připojeny dva ovládací prvky. Prvním je rotační kodér s axiálním potvrzovacím kontaktem. Ten slouží pro pohyb v menu a nastavení požadované hodnoty. Dále je připojeno tlačítko „ESCAPE“, po jehož stisku se dostaneme o jednu úroveň menu výše.



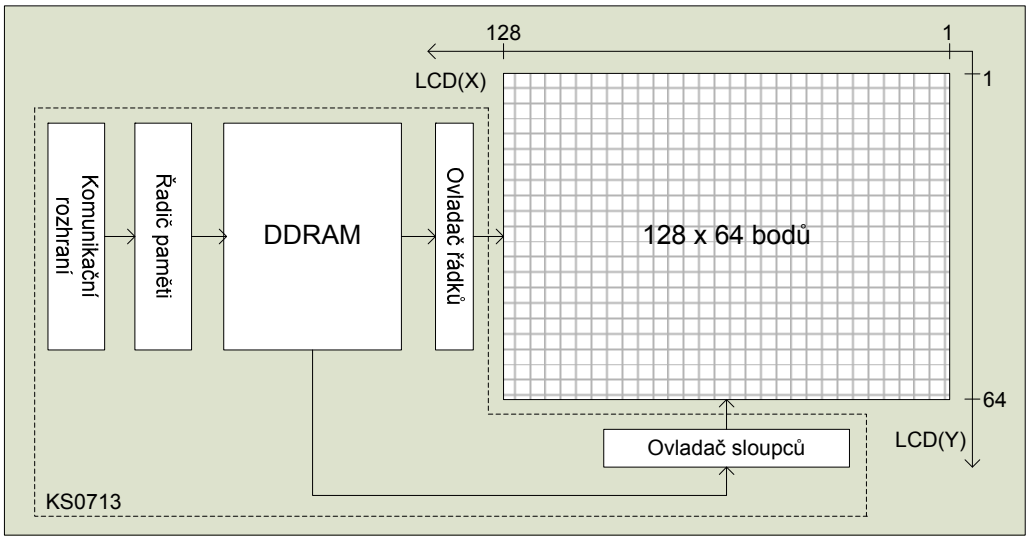
Obr. 20: Blokové schéma digitální části

Abychom mohli mikrokontroler programovat, resp. použít debugging, je vyvedeno sériové programovací rozhrání ICSP (In Circuit Serial Programming). Pro komunikaci s počítačem je vyvedena sběrnice USB. Dále jsou ke každé analogové části připojeny dva

signály. První z nich určuje polaritu výstupního napětí a tím i směr pohybu. Druhý signál má obdélníkový průběh se střídou 1:1 a frekvenci odpovídající nastavené rychlosti posuvu (1 – 500 Hz).

3.3.1 Grafický displej

Pro zobrazení ovládacích prvků a aktuálního nastavení mikroposuvu využívám plně grafický displej EA W128A-6X9 [8] s rozlišením 128x64 bodů. Displej má integrovaný řadič Samsung KS0713 [9] s vestavěnou RAM pamětí. Z následujícího obrázku plyne, že řadič KS0713 se skládá z komunikačního rozhraní, řadiče paměti, paměti DDRAM (Display Data RAM) a z ovladačů jednotlivých segmentů (řádků/sloupců).



Obr. 21: Zjednodušené blokové schéma displeje

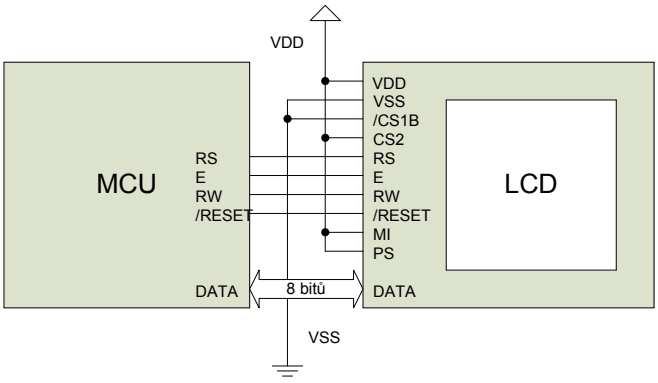
Komunikační rozhraní slouží pro komunikaci displeje s okolím, např. mikrokontrolerem. Jedná se o uživatelsky volitelné paralelní nebo sériové rozhraní s 3,3 V logikou. Dále následuje řadič displeje, který dekóduje příchozí instrukce, nastavuje pomocné registry nebo kopíruje příchozí data do DDRAM.

Paměť DDRAM uchovává informaci o tom, co má být zobrazeno na displeji. Na obrázku 22 je vidět, jakým způsobem je paměť rozvržena, resp., jak je v ní uložena informace o obraze. Paměť je rozdělena do 132 sloupců a 9 stránek, kde každá z prvních osmi stránek obsahuje osm řádků. Poslední stránka má pouze jeden řádek. Odtud plyne její velikost 8580 bitů (132 sloupců * (8 stránek po 8 řádcích + 1 stránka s 1 řádkem)). Protože rozlišení displeje je pouze 128*64 bodů, tak není celá paměť využita. Viditelná oblast je na obr. 22 vyznačena červenou čárou.

Adresa stránky P3,P2,P1,P0	DATA	Adresa RAM																Adresa řádku (HEX)	Souřadnice LCD(Y)
0,0,0,0	DB0					■				■				■	■	■		00	1
	DB1					■				■			■				■	01	2
	DB2					■				■			■					02	3
	DB3					■		■		■				■	■	■		03	4
	DB4					■				■							■	04	5
	DB5					■		■		■			■				■	05	6
	DB6						■		■					■	■	■		06	7
	DB7																	07	8
≀																		≀	≀
0,1,1,1	DB0																	38	57
	DB1																	39	58
	DB2																	3A	59
	DB3																	3B	60
	DB4																	3C	61
	DB5																	3D	62
	DB6																	3E	63
	DB7																	3F	64
1,0,0,0	DB0																	40	65
Adresa sloupce	ADC=0	83	82	81	80	7F	7E	7D	7C	7B	7A	~	5	4	3	2	1	0	
	ADC=1	0	1	2	3	4	5	6	7	8	9	~	7E	7F	80	81	82	83	
Souřadnice LCD(X)		132	131	130	129	128	127	126	125	124	123	~	6	5	4	3	2	1	

Obr. 22: Display Data RAM

Na obrázku 23 je vidět připojení displeje k mikrokontroleru. Komunikace mezi mikrokontrolerem a displejem probíhá po osmi datových a čtyřech řídících vodičích. Ostatní vodiče jsou napevno připojeny buď na napájecí napětí nebo na zem. Význam jednotlivých vodičů je uveden v tabulce 1.



Obr. 23: Schéma připojení displeje k mikrokontroleru

Protože chceme s displejem komunikovat po paralelní sběrnici, je na vstup PS přivedena logická 1. Stejnou logickou úroveň přivedeme i na vstup MI, protože požadujeme rozhraní kompatibilní s procesorem 6800 (nízká úroveň odpovídá procesoru 8080). Pokud bychom měli více displejů připojených na stejnou sběrnici, tak pomocí vstupů CS1B a CS2 vybíráme,

se kterým konkrétním displejem chceme komunikovat. V našem případě je připojen pouze jeden displej, tak vstup /CS1B připojíme napevno k zemi (nízká aktivní úroveň) a vstup CS2 na VDD (vysoká aktivní úroveň). Zbývající řídící vstupy RS, RW, E a /RESET jsou již ovládány mikrokontrolerem.

Tab. 2: Význam vodičů LCD displeje

Symbol	Směr dat	Popis
/CS1B	Vstup	Chip select 1
CS2	Vstup	Chip select 2
RS	Vstup	Výběr registru
E	Vstup	Povolovací vstup
RW	Vstup	Čtení / Zápis
/RESET	Vstup	Reset
MI	Vstup	Typ rozhraní
PS	Vstup	Paralelní / Sériová komunikace
DATA	Vstup / Výstup	8bitové datové slovo

Pokud je na vstupu RS logická jednička, znamená to, že 8bitové slovo přivedené na vstup DATA je nahráno do aktuální pozice zobrazovací paměti DDRAM. Pokud je přivedena logická nula, tak se jedná o ovládací příkaz. Význam některých příkazů je uveden v tabulce 3. Vstup RW přepíná směr osmibitové datové sběrnice DATA. Při vysoké úrovni jsou data čteny z displeje, při nízké úrovni mikrokontroler zapisuje data do displeje. Při sestupné hraně signálu E displej zpracuje přivedená data. Signál /RESET se používá pro inicializaci displeje.

Tab. 3: Základní příkazy LCD displeje

Instrukce	RS	RW	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0	Popis
read_dram	1	1	DATA								Vyčtení dat z aktuální pozice DDRAM
write_dram	1	0	DATA								Zápis dat na aktuální pozici DDRAM
read_status	0	1	BUSY	ADC	ON	RESET	0	0	0	0	Přečte stav displeje
disp_on	0	0	1	0	1	0	1	1	1	DON	DON = 1: Desplej zapnut, DON = 0: Displej vypnut
init_line	0	0	0	1	ST5	ST4	ST3	ST2	ST1	ST0	Nastavuje řádek DDRAM na kterém leží první řádek LCD(Y)
set_page	0	0	1	0	1	1	P3	P2	P1	P0	Vybere stránku paměti
set_col	0	0	0	0	0	1	Y7	Y6	Y5	Y4	MSB vybraného sloupce
	0	0	0	0	0	0	Y3	Y2	Y1	Y0	LSB vybraného sloupce
reset	0	0	1	1	1	0	0	0	1	0	softwarový reset

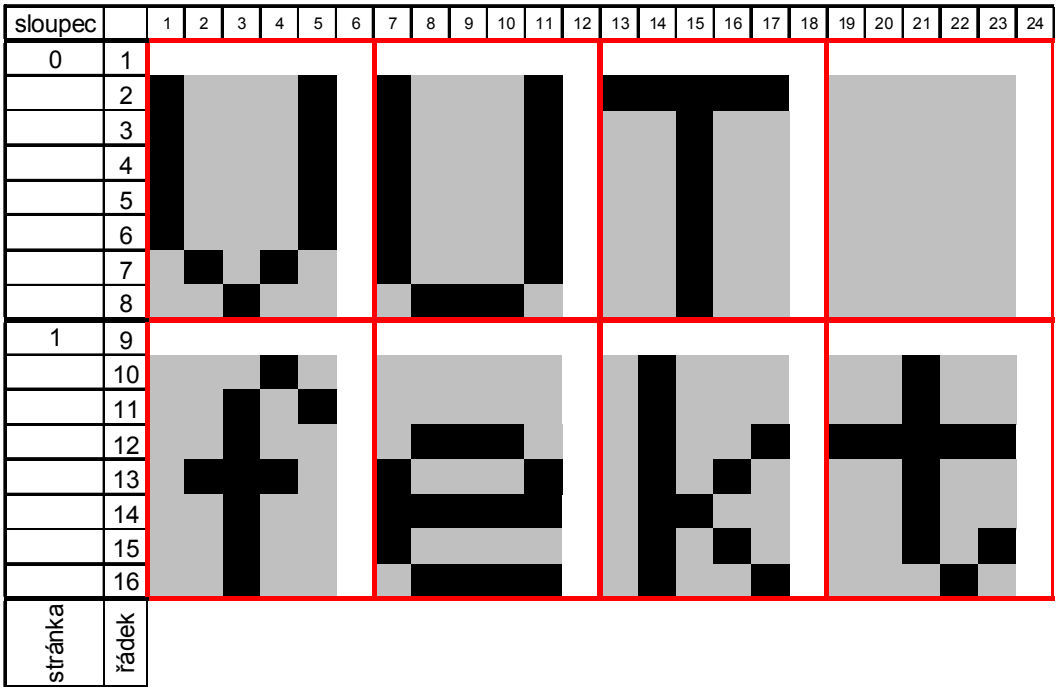
Inicializace displeje začíná nastavením signálu /RESET do nízké úrovně. V tomto stavu displej necháme po dobu ustálení napájecího napětí (výrobcem doporučená hodnota je 1 μs). Poté přivedeme na vstup /RESET vysokou úroveň. Tímto jsme provedli hardwarový reset displeje. Následně pomocí instrukce *reset* provedeme softwarový reset a čekáme, dokud

pomocí instrukce *read_status* nenačteme na čtvrtém bitu datové sběrnice DATA logickou 0. Nyní přeneseme uživatelské nastavení displeje, např. číslování sloupců (zprava doleva, nebo naopak) nebo číslování řádků (vzestupně nebo sestupně). Následuje instrukce pro zapnutí interního napěťového měniče a pauza minimálně 1 ms pro jeho stabilizaci. Další instrukcí zapneme interní napěťový regulátor a opět počkáme minimálně 1 ms na jeho stabilizaci. Následně zapneme interní napěťový sledovač a přeneseme nastavení kontrastu displeje. Protože obsah paměti DDRAM po připojení napájecího napětí není známa, resp. nabývá náhodných hodnot, tak provedeme její vymazání. Nyní můžeme pomocí instrukce *disp_on* displej zapnout a inicializace je skončena.

Pokud je displej zinicializovaný můžeme přistoupit k přenosu dat do DDRAM, resp. ke zobrazování informací na displeji. Zobrazitelná plocha displeje je rozdělena do 128 sloupců a 8 stránek, kde stránka banka má 8 řádku. Přenos do paměti probíhá tak, že vybereme, kterou stránku chceme používat (pomocí instrukce *set_page*), dále vybereme sloupec, který budeme používat (instrukce *set_col*), a nakonec data přeneseme pomocí instrukce *write_dram*. Těchto přenesených 8 bitů odpovídá 8 bodům zobrazených na displeji ve zvoleném sloupci a stránce.

Aby mohl displej zobrazovat znaky, je potřeba mít generátor znaků. Vzhledem k tomu, že radič displeje KS0713 jej neobsahuje, bylo nutné tento generátor naprogramovat v mikrokontroleru. Rozhodl jsem se používat znaky o velikosti 5x7 bodů. Aby byly znaky na displeji čitelné a nedocházelo ke spojení dvou sousedících znaků, je nad každý znak vložena horizontální mezera a napravo od znaku vertikální mezera. Jak je vidět z obrázku 24, jeden znak tedy zabírá plochu 6x8 bodů (plocha ohraničená červenou čárou), ale velikost jednoho znaku je maximálně 5x7 bodů (šedá plocha). Samotný znak je zobrazen černou barvou.

Abychom takto mohli znaky zobrazovat, musí být v paměti uloženo, jak jednotlivé znaky vypadají. Protože toto pole bude konstanta (v průběhu programu nebudeme měnit jeho hodnotu), může být uloženo v paměti programu a nemusí se pro něj tedy alokovat paměť RAM. Toto je obrovská výhoda, protože pokud spočítáme, že každý znak bude složen z 5 bajtů a budeme chtít generovat znaky 32 až 127 z ASCII tabulky, tak bude mít toto pole velikost 475 bajtů a vzhledem k tomu, že velikost paměti RAM u mikrokontroleru 18LF4550 je 2 kB, tak bychom tímto polem obsadili zhruba její čtvrtinu. Pokud je toto pole uloženo do paměti programu (ROM, resp. FLASH), tak to pro nás není již nijak omezující, protože její velikost je 32kB.



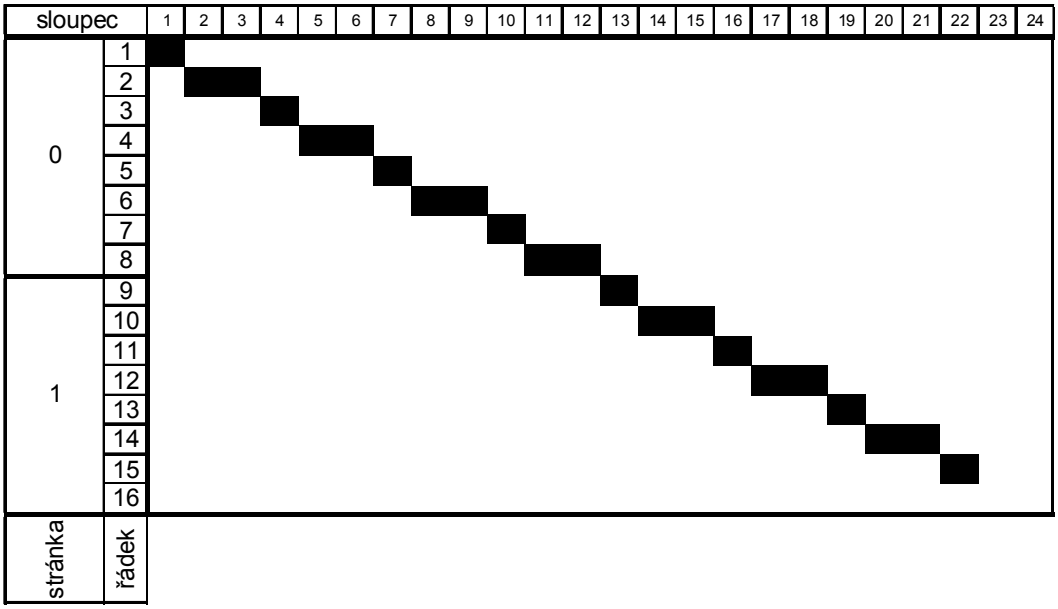
Obr. 24: Výstup generátoru znaků

Pro naše účely postačují velká a malá písmena, číslice a speciální znaky jako např. #, $, * atd. Z klasické ASCII tabulky tedy vyloučíme její rozšířenou část (znaky 128 – 255) a řídící znaky (0 – 31). Zbude tedy pole znaků s ASCII kódy 32 až 127. Každý znak v tomto poli je reprezentován pěti bajty. Potom tedy velikost tohoto pole bude (127-32)*5=475 bajtů. Víme tedy, že první sloupec znaku s ASCII kódem 32 (mezera) má pozici v poli 0, druhý sloupec má pozici 1, … , pátý sloupec má pozici 5. Dále následuje znak s ASCII kódem 33 (vykřičník). Pozice prvního sloupce tohoto znaku je v poli na šestém místě, druhý sloupec je na pozici 7 atd. Máme-li tedy v paměti uložené takovéto pole znaků, můžeme napsat funkci která pro libovolný znak ASCII tabulky v rozmezí 32 – 127 nakopíruje data do DDRAM tak, aby se zvolený znak zobrazil na displeji.

Pro kreslení horizontálních a vertikálních čar na displeji můžeme rovnou data zapisovat do paměti DDRAM a to tak, že pro horizontální čáru máme konstantní stránku paměti i data představující jednotlivé řádky ve zvoleném sloupci a měníme pouze sloupce v paměti DDRAM. Pro vertikální čáru naopak měníme stránky paměti, resp. jednotlivé řádky ve zvoleném sloupci, ale sloupec se nemění.

Pro šikmé čáry používám Bresenhamův přímkový algoritmus [10], který zakreslí vektorově zadanou přímku do rastrového pole jednotlivých segmentů LCD displeje. Výchozí informací pro tento algoritmus je počáteční a koncový bod úsečky, kterou chceme vykreslit. Mezi těmito krajními body jsou body mřížky vyplňovány tak, že se dotýkají buď společnou

hranou, anebo společným vrcholem segmentů. Na obrázku 25 je zobrazena úsečka s počátečním bodem [1;1] a koncovým bodem [22;15] nakreslená pomocí Bresenhamova algoritmu.



Obr. 25: Bresenhamův přímkový algoritmus

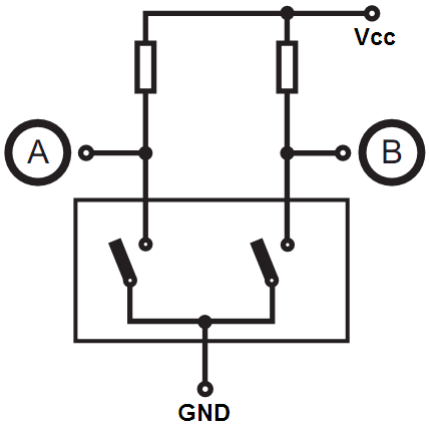
Protože při kreslení úseček by při přímém zápisu do paměti docházelo k přepsání celého aktuálně zvoleného sloupce v rámci jedné stránky paměti a nebylo by možné nakreslit třeba křížení čar, je nutné nejdříve tento sloupec z pamětí DDRAM přečíst. Následně, podle zvoleného módu kreslení (PIXEL_ON, PIXEL_OFF, PIXEL_INVERT) a informaci o původní hodnotě aktuálního sloupce, rozhodnout, jaká hodnota má být uložena zpět do paměti.

3.3.2 Rotační kodér

Jako ovládací prvek pro pohyb v menu byl zvolen inkrementální rotační kodér. Konkrétně se jedná o mechanický rotační kodér s axiálním potvrzovacím kontaktem P-RE30S [11].

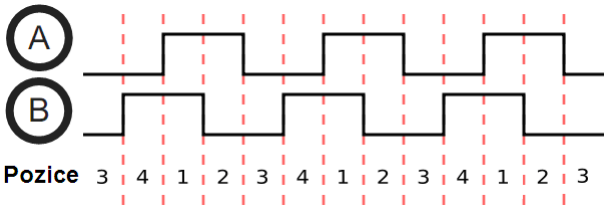
Obr. 26: Rotační kodér P-RE30S

Princip tohoto ovládacího prvku je v postupném spínání a rozepínání dvou fázově posunutých spínačů. Schéma zapojení takovéhoto kodéru je na obrázku 27. Pokud je spínač rozepnutý, tak se na výstupu objeví napájecí napětí Vcc. Naopak, pokud je spínač sepnut, pak je výstup uzemněn. Protože se jedná o mechanické spínače, tak vlivem různých přechodových dějů dochází k jejich zakmitávání. Podle datasheetu výrobce je maximální délka zákmitu při dodržení specifikací (proud kontaktem, teplota, relativní vlhkost) 5 ms. Toto je největší nevýhoda oproti optickým rotačním kodérům, které jsou ovšem několikanásobně dražší.



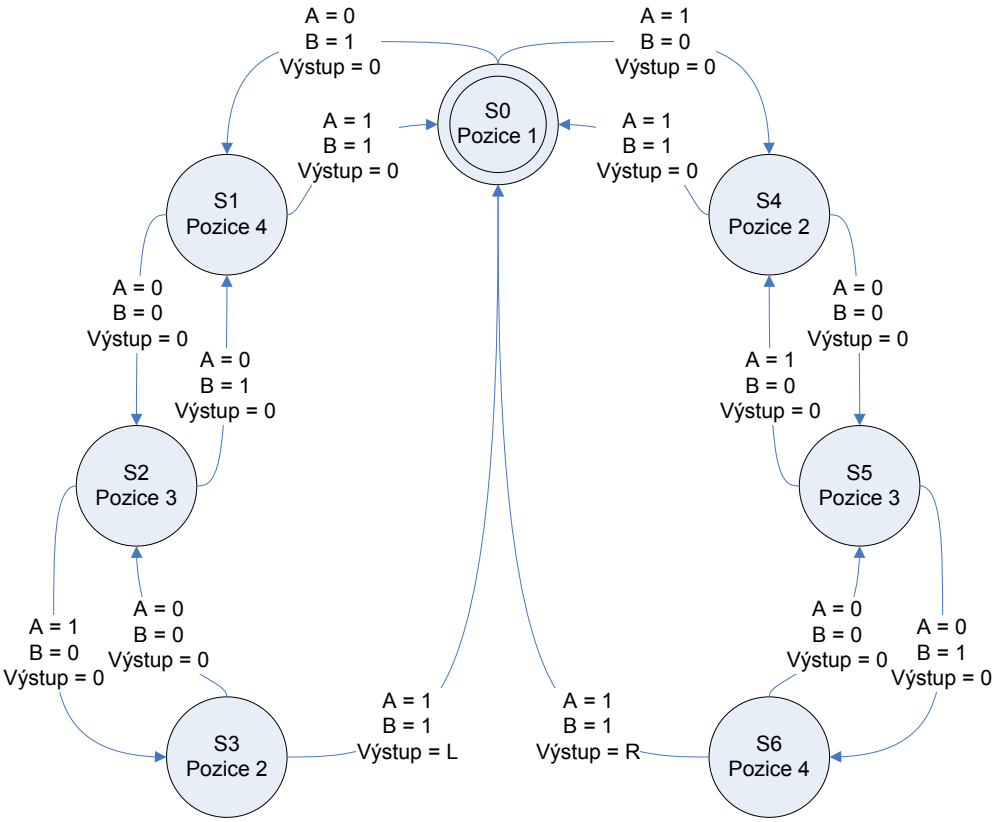
Obr. 27: Schéma zapojení rotačního kodéru

Mnou použitý rotační kodér má 30 kroků na jednu otáčku, tj. 30 klidových stavů, ve kterých se hodnota výstupu A a B nemění. Protože v tomto stavu jsou oba spínače rozepnuty, tak na výstupech A a B je vysoká úroveň. Tomuto stavu odpovídá pozice 1 na obrázku 28. Pokud otočíme kodérem ve směru hodinových ručiček, pak hodnoty výstupů prochází přes pozice 2, 3 a 4 opět do klidové pozice 1. Otáčíme-li v protisměru pohybu hodinových ručiček, pak výstupy prochází pozicemi 4, 3, 2 a končí opět na klidové pozici 1.



Obr. 28: Průběhy výstupů rotačního kodéru

Existuje mnoho způsobů jak vyhodnocovat pohyb rotačního kodéru. Lze například použít obvody firmy LSI/CSI, která se specializuje na výrobu obvodů vyhodnocujících inkrementální kodéry. Já jsem se rozhodnul použít softwarové vyhodnocování pohybu. K tomuto účelu jsem naprogramoval stavový automat. Stavový diagram tohoto automatu je na obrázku 29.



Obr. 29: Stavový diagram

Po spuštění programu je automat nastaven na stav S0, kodér je v klidovém stavu, tj. výstup A i B je v logické 1. Pokud pootočíme kodérem ve směru pohybu hodinových ručiček, výstup A zůstává v logické 1, ale výstup B mění svojí hodnotu na logickou 0. Tímto se dostáváme se do pozice 2, resp. stavu S4. Dále mění svojí hodnotu na nízkou úroveň i výstup A, a tímto se dostáváme do stavu S5. Následně jde výstup B do vysoké úrovně a stav se mění na S6. Nakonec jde do vysoké úrovně i výstup A, a tím přechází kodér opět do klidového stavu S0. Touto sekvencí bylo detekováno otočení kodéru o jeden krok ve směru otáčení hodinových ručiček. Detekce opačného směru je analogická.

Výhodou takovéhoto stavového automatu je eliminace vlivu zákmitů spínačů, protože při změně pozice o 1 dochází k zákmitu pouze jednoho spínače, který způsobí, po dobu trvání přechodového děje, přeskakování mezi dvěma sousedícími stavy. Po odeznění zákmitu se nastaví takový stav, který odpovídá aktuální hodnotě výstupu kodéru. V tomto stavu automat čeká na posun o další pozici.

3.3.3 Časovač

Pro generování signálu ovládajícího mikroposuv využívám tři 16bitové časovače. Tři proto, že generuji tři nezávislé kmitočty – 3 osy posuvu. Protože kmitočet musí být plně přeladitelný v intervalu 1 Hz – 500 Hz s krokem 1 Hz, je nutné generovat přerušení s frekvencí 2 Hz – 1 kHz (aby měl výstupní signál střídu 1:1 a frekvenci 1 Hz, resp. perioda signálu byla 1 s, musí být výstup 0,5 s v logické 0 a 0,5 s v logické 1 → frekvence s jakou je vyvoláno přerušení je tedy dvojnásobkem požadovaného výstupního kmitočtu).

Takt oscilátoru je $f_{OSC} = 16\,MHz$. Mikrokontroler potřebuje 4 takty oscilátoru, aby provedl jedno-cyklovou instrukci. Můžeme tedy vypočítat, že doba jakou bude mikrokontroler provádět jednu instrukci je:

$$T_{INS} = \frac{4}{f_{OSC}} = \frac{4}{16 \cdot 10^6} = 250\,ns \tag{5}$$

Čítače TIMER1, TIMER2 a TIMER3 jsou nastaveny tak, aby zvýšily svojí hodnotu o jedničku s každým instrukčním cyklem, tedy každých 250 ns. Pokud chceme generovat signál o kmitočtu 500 Hz, resp. o periodě 2 ms, pak potřebujeme vyvolat přerušení po 1 ms. Pokud nastavíme čítač tak, aby při přetečení z 0xFFFFh na 0x0000h vyvolal přerušení a nahrajeme do něj vhodně vypočítanou hodnotu, pak nám bude čítač generovat přerušení právě po 1 ms. Počet instrukčních cyklů, které musí čítač načítat, vypočítáme z následujícího vztahu:

$$x = \frac{T_{INT}}{T_{INS}}, \tag{6}$$

kde x je počet instrukčních cyklů, T_{INT} je perioda, s jakou se má vyvolávat přerušení a T_{INS} je doba instrukčního cyklu.

Dosazením do vzorce (6) dostáváme:

$$x = \frac{1 \cdot 10^{-3}}{250 \cdot 10^{-9}} = 4000 \tag{7}$$

Protože čítač čítá od 0x0000h do 0xFFFFh, tak musíme hodnotu vypočtenou v (7) odečíst od maximální hodnoty čítače, tj. 0xFFFF (v dekadické soustavě 65535). Po odečtení získáváme přímo hodnotu, na kterou musíme přednastavit čítač.

$$TMR = 65535 - x = 65535 - 4000 = 61535 \tag{8}$$

Podle výše uvedených vzorců byl vytvořen vzorec (9), podle něhož se v mikrokontroleru vypočítávají hodnoty, na které se přednastavují jednotlivé čítače.

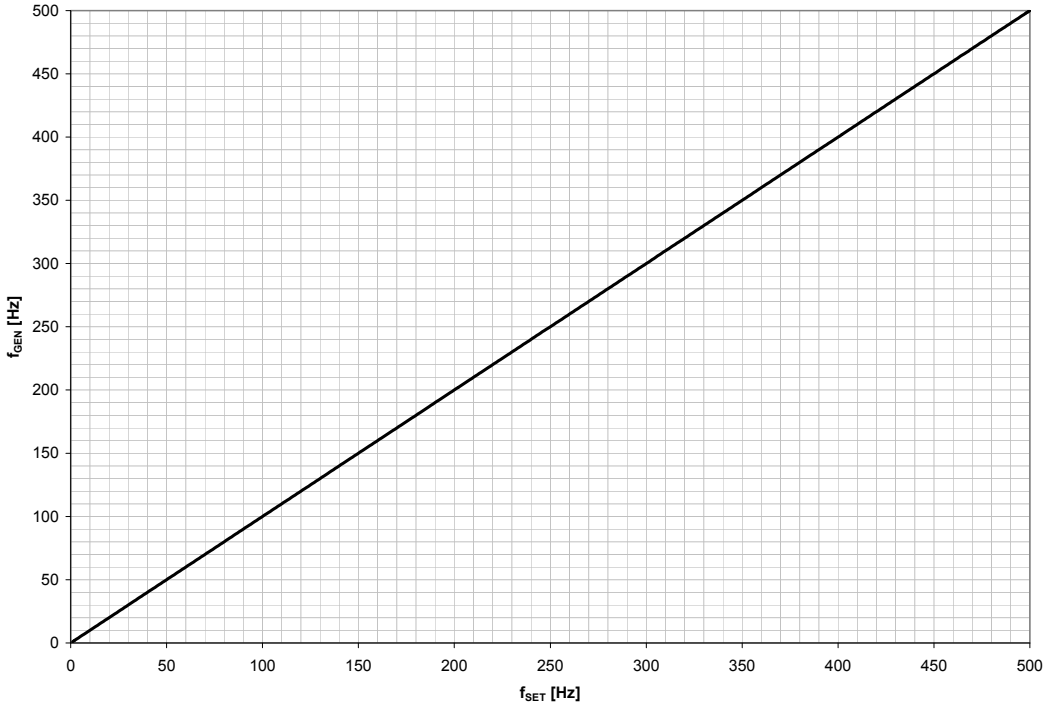
$$TMR = 65535 - \frac{f_{OSC}}{8 \cdot PRESCALE \cdot f_{SET}}, \tag{9}$$

kde f_{OSC} je frekvence oscilátoru, f_{SET} je nastavená výstupní frekvence a PRESCALE je hodnota programovatelné předděličky.

Protože hodnota čítače je 16bitové číslo, tak nejdelší možná doba, jakou čítač bude čítat do přetečení je dána vztahem:

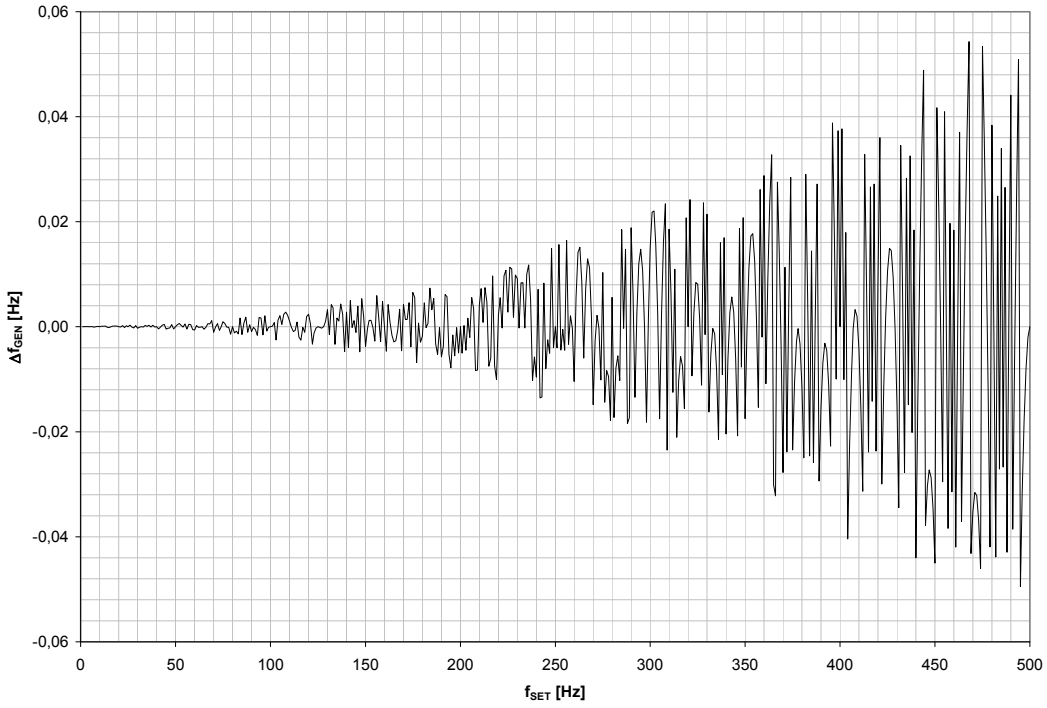
$$T_{MAX} = 65535 \cdot T_{INS} = 65535 \cdot 250 \cdot 10^{-9} \approx 16{,}38\,ms \tag{10}$$

Pokud bychom překlápěli výstup každých 16,38 ms, pak by byla výstupní frekvence 30,5 Hz. Proto, pokud potřebujeme generovat výstupní signál o frekvenci nižší než 31 Hz, je zapotřebí použít programovatelnou předděličku. Ta definuje kolik instrukčních cyklů (1, 2, 4, 8, …) je potřeba ke zvýšení hodnoty čítače o 1.

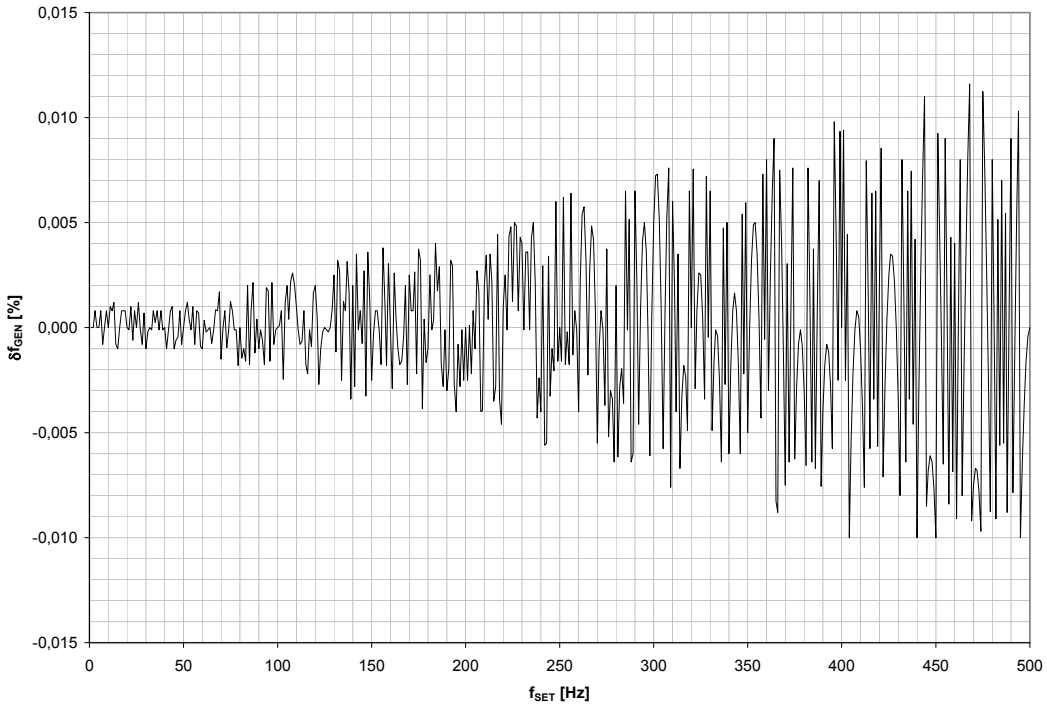


Obr. 30: Graf závislosti generované frekvence na nastavené frekvenci

Protože hodnota čítače TMR je celé 16bitové číslo, tak dochází při výpočtu dle (9) k chybě díky zaokrouhlování. Následující dva grafy ukazují absolutní a relativní chybu generované frekvence (oproti nastavené). Z grafů je patrné, že chyba se vzrůstající nastavenou frekvencí roste. Hodnota této chyby je dána pouze zbytkem po dělení dvou celých čísel.



Obr. 31: Graf závislosti absolutní chyby generátoru na nastavené frekvenci



Obr. 32: Graf závislosti relativní chyby generátoru na nastavené frekvenci

3.3.4 Microchip USB Firmware Framework

Microchip USB Firmware Framework [12] je knihovna, která se používá k vytvoření USB aplikací. Jedná se v podstatě o projekt obsahující veškerý potřebný kód pro obsluhu USB sběrnice a dále prostor pro uživatelský kód, včetně obsluhy přerušení. Tento projekt se skládá z následujících podadresářů a souborů:

Podadresáře:

- *_output* - místo pro uložení zkompilovaných souborů
- *autofiles* - obsahuje globální konfigurace a deskriptory USB
- *system* - obsahuje Microchip USB firmware
- *user* - místo pro uložení uživatelského kódu

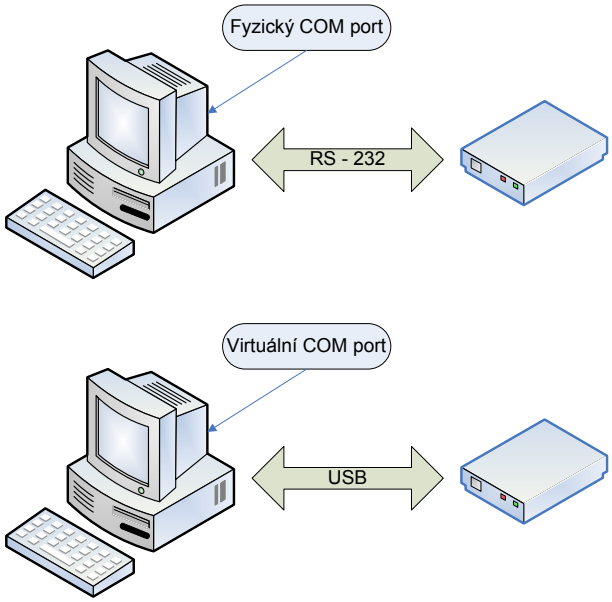
Soubory:

- *CleanUp.bat* - dávka pro smazání všech zkompilovaných souborů
- *io_cfg.h* – hlavičkový soubor mapující jména pinů a konfigurace portů
- *main.c* - obsahuje funkci *main()*

Microchip USB firmware obsahuje sadu modulů, které dohromady pokrývají většinu práce pro implementaci USB komunikace. Na tomto místě uvádím přehled jednotlivých modulů frameworku a jejich hlavičkových souborů s jejich stručným popisem.

- *main.c* - při spuštění funkce *main()* se volá funkce *InitializeSystem()*. Je to centrální funkce pro inicializaci a všechny potřebné inicializační rutiny jsou volány odtud. Dále následuje nekonečná smyčka programu *while(1)*. V této smyčce jsou postupně volány funkce *USBTasks()* a *ProcessIO()*. Funkce *USBTasks()* slouží k obsluze USB úkolů, a to prostřednictvím funkce *USBDriverService()* z modulu *usbdrv.c*. Tato funkce obsluhuje všechna USB přerušení a volí příslušné obslužné rutiny. Funkce *ProcessIO()* z modulu *user.c* obsahuje uživatelskou část programu.
- *autofiles / usbcfg.h* - v tomto souboru se provádí konfigurace USB rozhraní.
- *autofiles / usbdsc.h a .c* - modul obsahující informace o USB deskriptorech.
- *system / typedefs.h* - hlavičkový soubor obsahující definice použitých datových typů.

- *system / interrupt / interrupt.c a .h* - tento modul obsluhuje přerušení. Je zde definováno makro *mEnableInterrupt()*, které povoluje globální přerušení. Vlastní obsluha přerušení je pak implementována ve funkcích *low_isr()* a *high_isr()*.
- *system / usb / usb.h* - centrální hlavičkový soubor frameworku. Zahrnutím tohoto souboru do projektu jsou všechny globální proměnné a funkce šířeny do ostatních modulů.
- *system / usb / class / cdc / cdc.c a .h* - modul obsahuje implementaci funkcí rozhraní zařízení třídy CDC (Communication Device Class). Tato třída, jak naznačuje obrázek 33, umožňuje nahrazení fyzického sériového portu portem virtuálním.
- *system / usb / usbdefs / usbdefs_ep0_buff.h* - tento soubor definuje strukturu setup paketu *CTRL_TRF_SETUP* a paketu odpovědi na řídící přenos *CTRL_TRF_DATA*.
- *system / usb / usbdefs / usbdefs_std_dsc.h* - soubor obsahující definice standardních deskriptorů USB_DEV_DSC, USB_CFG_DSC, USB_INTF_DSC a USB_EP_DSC a jejich parametrů.
- *system / usb / usbmap.h a .c* - modul představující USB paměťový manažer.
- *system / usb / usbdrv / usbdrv.c a .h* - tento modul je základem celého USB. Nejdůležitější funkcí je *USBDriverService()*, která se stará o všechna USB přerušení a volá ostatní funkce postupně tak, jak zařízení prochází jednotlivými stavy enumerace.
- *system / usb / usb9 / usb9.c a .h* - modul obsluhuje standardní USB žádosti přicházející přes endpoint 0.
- *system / usb / usbctrltrf / usbctrltrf.h a .c* - tento modul řídí přenosy a poskytuje služby všem implementovaným třídám (Human Interface Device, Communication Device Class, Mass Storage).



Obr. 33: Emulace RS-232 pomocí USB

Pro komunikaci s počítačem po USB sběrnici využívám třídu CDC (Communication Device Class) [13]. Pomocí této třídy, resp. pomocí serial emulation modelu, který je v této třídě definován, je možné nahradit dnes na většině počítačů se již nevyskytující sériový port. Pokud je na USB sběrnici připojeno CDC zařízení, dojde po načtení ovladače k vytvoření virtuálního sériového portu. Výhoda využití emulace RS-232 na USB sběrnici spočívá v jednoduchosti pasní nových a možnosti použití stávajících aplikací bez nutnosti jakýchkoliv změn. Díky implementaci třídy CDC v Microchip USB Firmware Frameworku je poměrně jednoduchá i aplikační část na straně mikrokontroleru. Detailnější popis použití této třídy je v aplikačním listu výrobce AN956 [14]. Za účelem emulace sériového portu je v modulu *cdc.c* vytvořena sada následujících funkcí:

- *void putrsUSBUSART(const rom char *data)*

  Funkce slouží pro odeslání řetězce z paměti programu. Vstupem je ukazatel na řetězec umístěný v programové paměti ukončený znakem NULL. Pokud není znak NULL nalezen, tak je odesláno pouze prvních 255 znaků řetězce.

- *void putsUSBUSART(char *data)*

  Funkce slouží pro odeslání řetězce z paměti dat. Vstupem je ukazatel na řetězec umístěný v datové paměti ukončený znakem NULL. Pokud není znak NULL nalezen, je odesláno pouze prvních 255 znaků řetězce.

- *void mUSBUSARTTxRom(rom byte *pData, byte len)*

  Makro sloužící k odesílání dat z programové paměti. Vstupem je ukazatel na první odesílaný bajt v programové paměti a délka, resp. počet odesílaných bajtů.

- *void mUSBUSARTTxRam(byte *pData, byte len)*

  Makro sloužící k odesílání dat z datové paměti. Vstupem je ukazatel na první odesílaný bajt v datové paměti a délka, resp. počet odesílaných bajtů.

- *BOOL mUSBUSARTIsTxTrfReady(void)*

  Makro sloužící ke kontrole, zda je třída CDC připravena k odeslání dat. Pokud je předešlý přenos ukončen a systém je schopen přenést další data, tak je návratová hodnota 1, v opačném případě 0. Před každým voláním funkce nebo makra odesílající data je nutné provést tuto kontrolu.

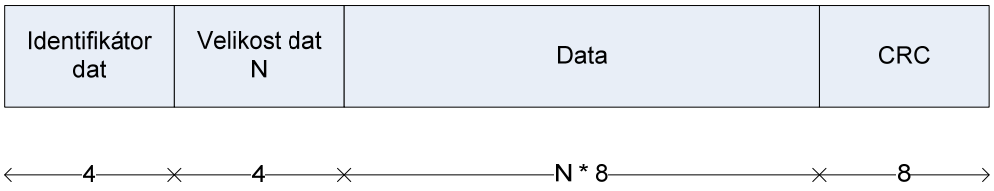
- *byte getsUSBUSART(char *buffer, byte len)*

  Funkce kopírující příchozí řetězec do paměti na pozici ukazatele *buffer*. *len* určuje očekávanou délku řetězce. Pokud je přijatý řetězec delší než *len*, pak je zkopírovaná pouze část o velikosti *len*. Návratová hodnota funkce je počet překopírovaných bajtů. Pokud nejsou žádná data k dispozici, funkce vrací 0.

3.3.5 Přenosový protokol

Na obrázku 34 je zobrazena struktura přenášeného paketu. Ten se skládá z identifikátoru dat, velikosti dat, dat samotných a kontrolního součtu. Význam identifikátorů je uvedený v tabulce 4. Další částí je velikost přenášených dat v bajtech. Následují data samotná. Paket končí kontrolním součtem všech předchozích částí.

Po přijetí paketu dojde k jeho kontrole pomoci CRC [15], a pokud je kontrolní součet shodný s přijatým kontrolním součtem, tak je přenos vyhodnocený jako bezchybný. Následně je zpět odeslán paket se stejným identifikátorem dat a s nulovou velikostí dat. Tímto způsobem je potvrzen příjem dat.



Obr. 34: Struktura paketu a jeho velikost v bitech

Tabulka 4: Význam identifikátoru dat

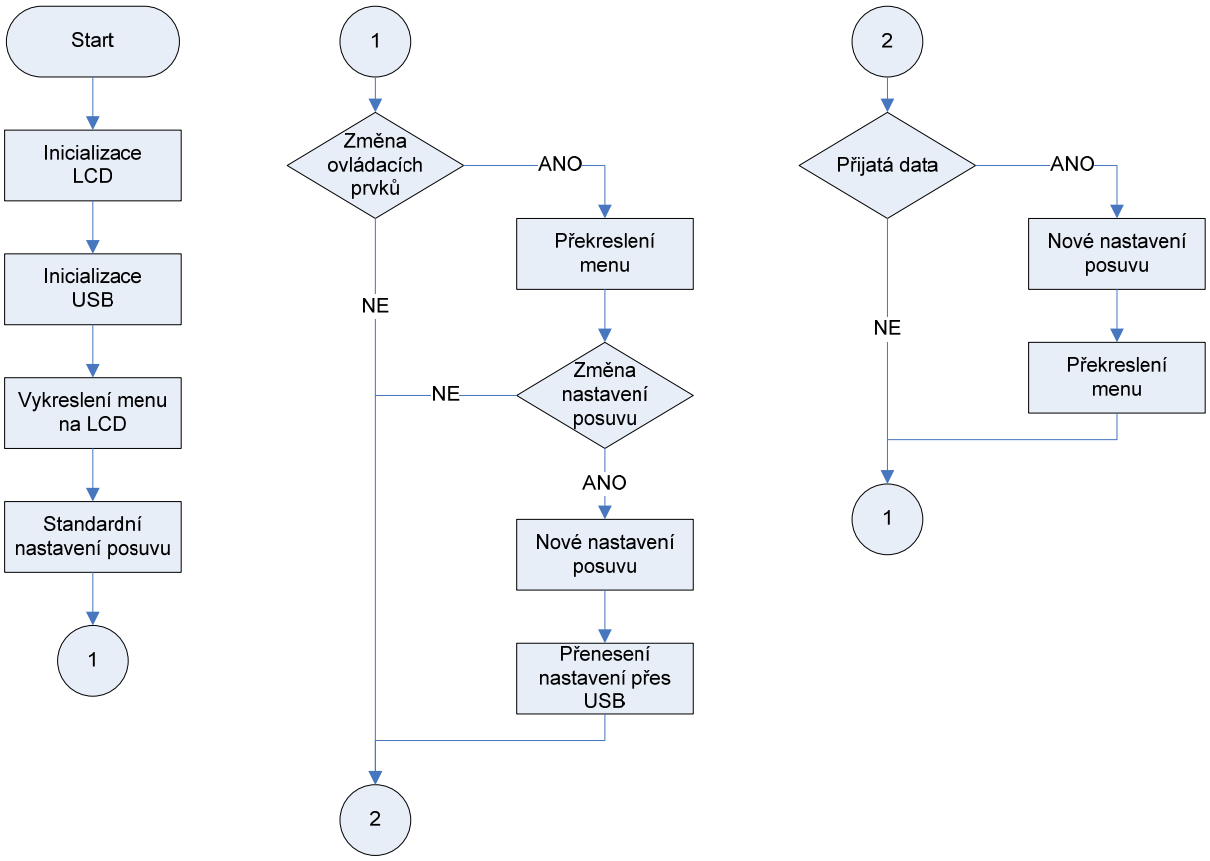
Identifikátor dat	Význam přenášených dat
1	Osa X - Nastavení rychlosti
2	Osa X - Počet kroků
3	Osa X - Směr
4	Osa X - Spuštěno
5	Osa Y - Nastavení rychlosti
6	Osa Y - Počet kroků
7	Osa Y - Směr
8	Osa Y - Spuštěno
9	Osa Z - Nastavení rychlosti
10	Osa Z - Počet kroků
11	Osa Z - Směr
12	Osa Z - Spuštěno

Z výše uvedené struktury vyplývá, že přenos dat je zatížen poměrně velkou režií. To je způsobeno univerzálností takovéhoto protokolu, kdy je možné prakticky bez změny jeho struktury přenášet další data (nastavení). Dále jistou nadbytečností, protože data přenášená po USB pomocí hromadných přenosů (Bulk-Transfer), kam spadá třída CDC, jsou již kontrolovaná pomocí 16bitového CRC a potvrzovaná pomocí handshake paketu. Uvedený protokol je tedy spíše ukázkou toho, jak vypadá potvrzovaná komunikace zabezpečená kontrolními součty. Protože je po sběrnici přenášeno pouze nastavení mikroposuvu a požadovaná rychlost je tedy maximálně v jednotkách bajtů za sekundu, tak zvýšená režie tohoto protokolu není závadou.

3.3.6 Program pro mikrokontroler

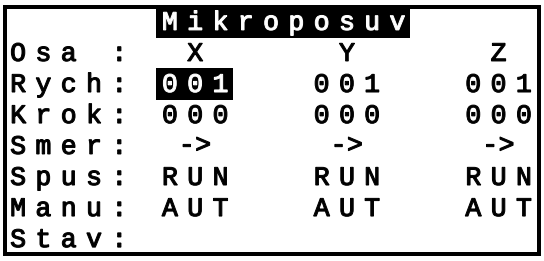
Program pro mikrokontroler je psán v jazyce C. K programování využívám prostředí MPLAB IDE [16] a kompilátor MPLAB C18 [17]. Pro komunikaci po USB sběrnici je využit Microchip USB Firmware Framework.

Na obr. 35 je zobrazen vývojový diagram k hlavnímu programu. Po spuštění programu dojde k inicializace LCD displeje a USB sběrnice. Následně je na displeji vykresleno menu (obr. 36) a je nahráno standardní nastavení. Dále následuje nekonečná smyčka, ve které se provádí kontrola změny ovládacích prvků (rotační kodér a tlačítko) a příjmu dat z počítače.



Obr. 35: Vývojový diagram hlavního programu

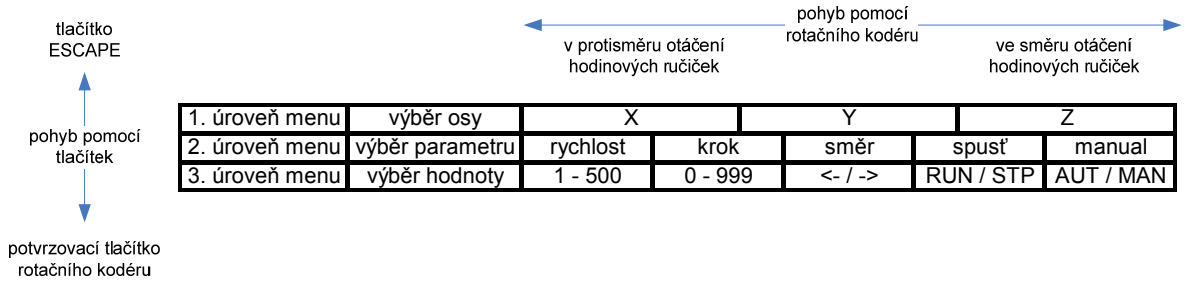
Pokud dojde ke změně ovládacích prvků, dojde k překreslení menu a dále je vyhodnoceno, zda se jedná pouze o změnu aktuální polohy v menu, nebo o změnu hodnoty položky v menu. Pokud se jedná o změnu hodnoty (rychlost, počet kroků, směr, atd.), je aplikováno nové nastavení.



Obr. 36: Menu zařízení ovládajícího mikroposuv

Pohyb v menu je znázorněn na obrázku 37. Aktuálně vybraná hodnota v menu je zvýrazněná inverzním písmem (obr. 36). Pokud jsme v první úrovni menu, tak se otáčením rotačního kodérů přepínáme mezi jednotlivými osami. Stisknutím potvrzovacího tlačítka potvrdíme výběr osy a nyní vybíráme parametr, který chceme pro konkrétní osu měnit. Po

dalším stisknutí potvrzovacího tlačítka můžeme nastavovat konkrétní hodnotu pro vybraný parametr. Stlačením tlačítka ESCAPE se přesuneme o jednu úroveň menu výše.



1. úroveň menu	výběr osy	X		Y		Z
2. úroveň menu	výběr parametru	rychlost	krok	směr	spusť	manual
3. úroveň menu	výběr hodnoty	1 - 500	0 - 999	<- / ->	RUN / STP	AUT / MAN

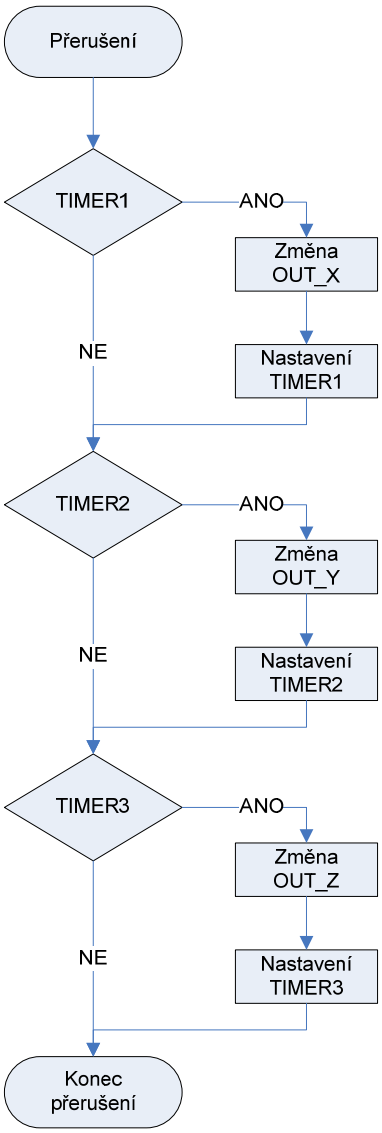
Obr. 37: Pohyb v menu

Význam jednotlivých parametrů:

- Rychlost – nastavuje rychlost mikroposuvu, resp. frekvenci generovaného signálu. Může nabývat hodnot od 1 – 500 Hz s krokem 1 Hz.
- Krok – nastavuje počet kroků mikroposuvu, resp. počet generovaných impulsů. Může nabývat hodnot od 1 – 998 kroků. Pokud je nastavena 0, pak tento parametr není brán v úvahu a impulsy se generují donekonečna, resp. do manuálního zastavení.
- Směr – přepíná směr posuvu, resp. polaritu výstupního napětí.
- Spusť – Při potvrzení volby RUN dojde ke spuštění posuvu pro aktuálně zvolenou osu. Hodnota parametru se nyní nastaví na STP (Stop). Pokud dojde k potvrzení volby STP, pak je posuv zastaven.
- Manuál – standardně je tento parametr nastaven na AUT. Pokud jej přepneme na MAN, ovládáme posuv rotačním kodérem, tj. pokud kodérem otočíme o jeden krok, tak se na výstupu vygeneruje jeden impuls. Polarita výstupního napětí odpovídá směru otáčení kodéru.

Pokud dojde k příjmu dat z počítače, je vypočítán jejich kontrolní součet, a pokud souhlasí s přijatým kontrolním součtem, tak je aktualizováno nastavení posuvu. Následně je překresleno menu s novými hodnotami jednotlivých parametrů.

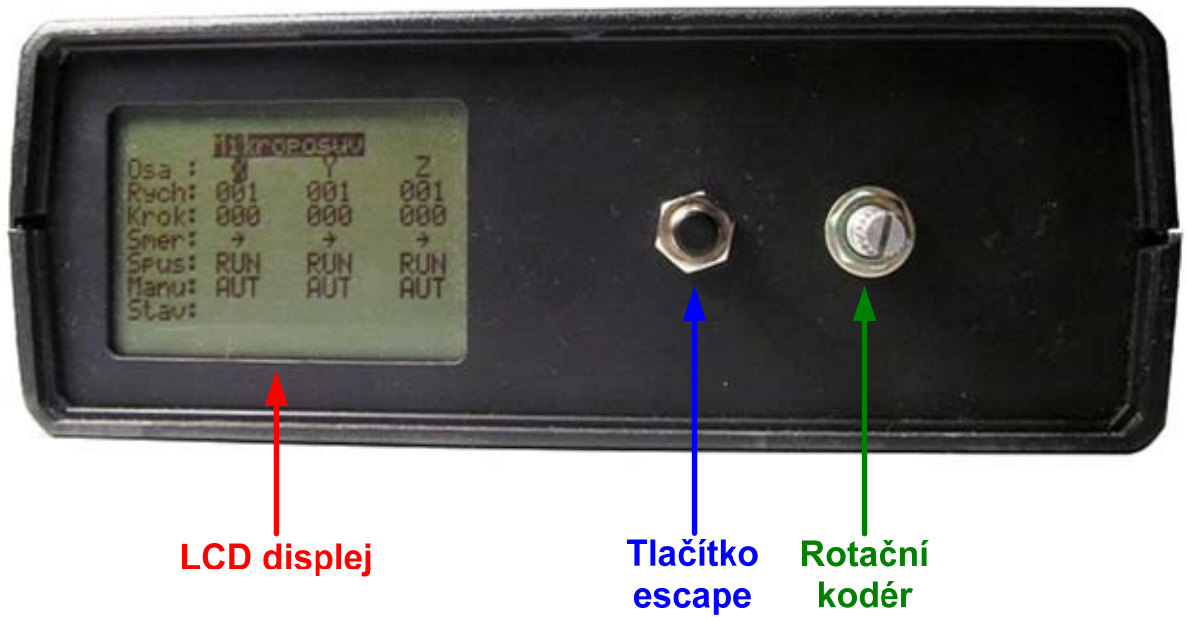
Na obrázku 38 je vývojový diagram pro rutinu přerušení. Pokud dojde k přetečení některého z časovačů, dojde ke změně hodnoty na příslušném výstupu a časovač je přednastaven na hodnotu vypočtenou dle (9).



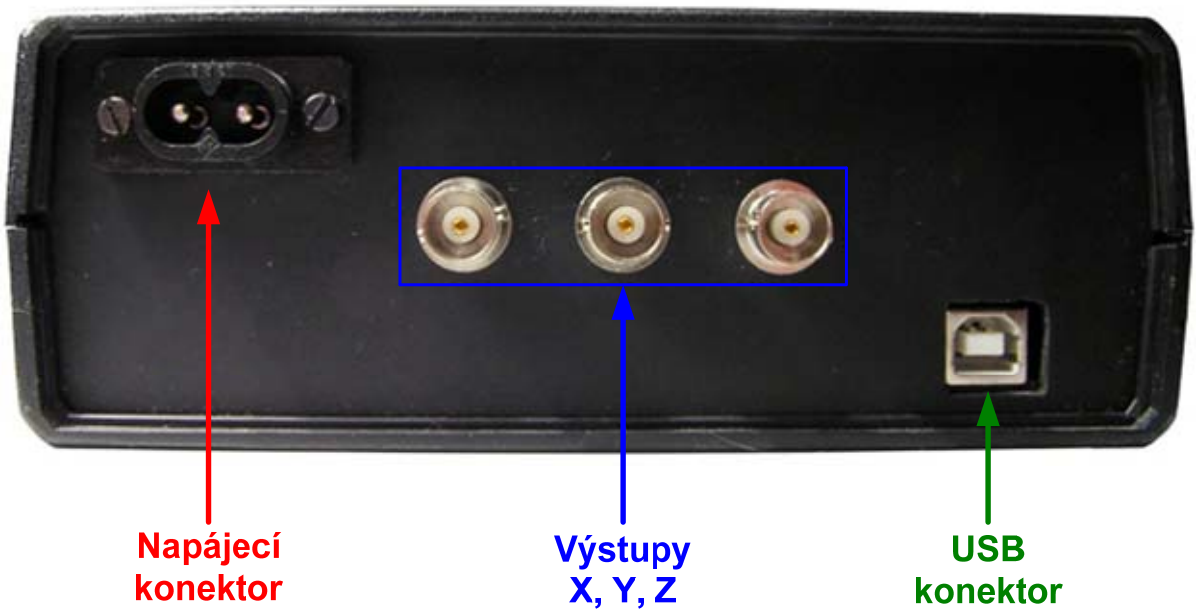
Obr. 38: Vývojový diagram rutiny přerušení

3.4 Praktická realizace a mechanické uspořádání

Zařízení je umístěno do krabičky o rozměrech 159x138x59 mm. V předním panelu je vyfrézován otvor pro LCD displej, tlačítko escape a pro rotační kodér. Na zadní straně je síťová zásuvka pro připojení napájecího napětí, konektor USB pro komunikaci s počítačem a tři výstupy s BNC konektory.

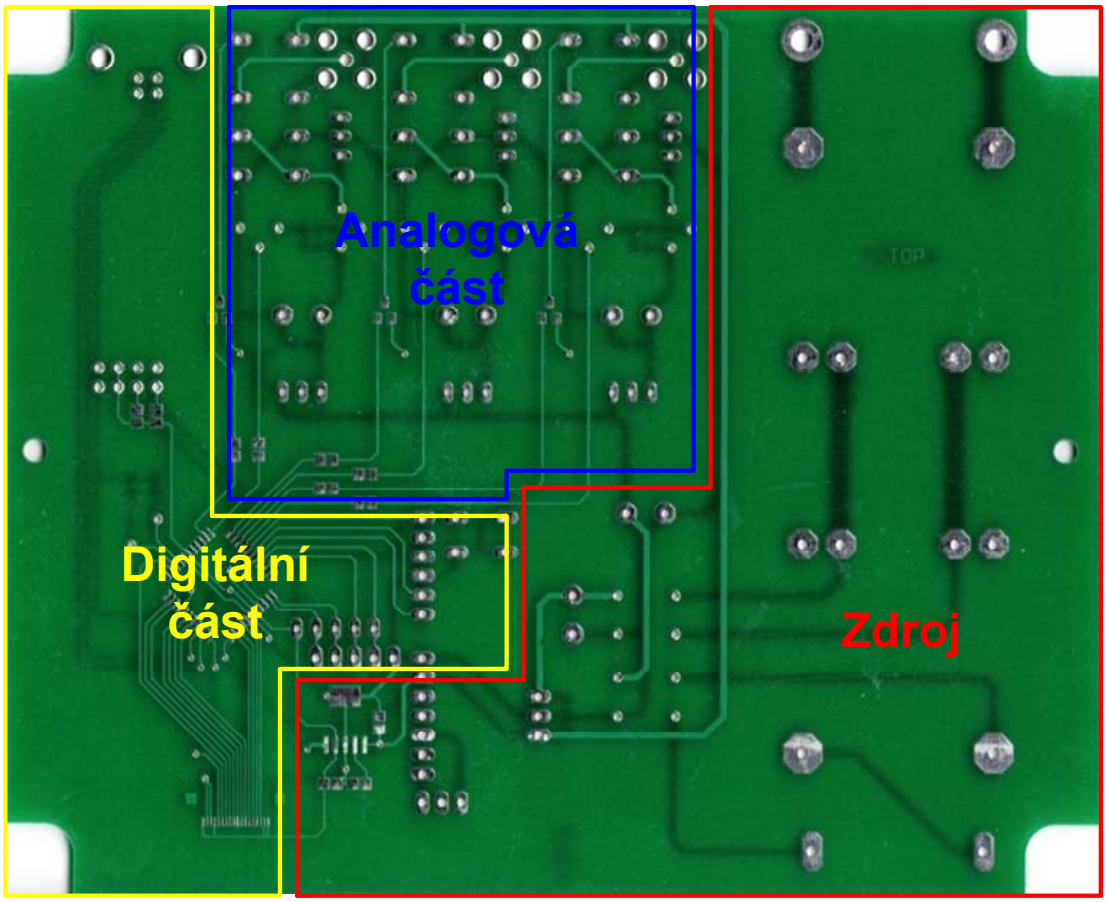


Obr. 39: Přední pohled



Obr. 40: Zadní pohled

Pro zařízení byla navrhnuta a zhotovena oboustranná deska plošných spojů s pokovenými otvory. Na povrchu DPS je sítotiskem nanesena teplem vytvrzovaná nepájivá maska. Na obrázku 41 je vidět rozložení jednotlivých bloků na DPS.



Obr. 41: Rozmístění na DPS

Na obr. 42 je celkový pohled na zařízení. V pravé části se nachází zdrojová část (na obr. 41 označená červeně) obsahující vstupní svorky pro připojení síťového napětí, dvojici transformátorů a můstkových usměrňovačů, filtrační kondenzátory a v neposlední řadě dvojici stabilizátorů. V levém spodním rohu je umístěna digitální část (na obr. 41 označená žlutě), jejíž srdcem je mikrokontroler. Dále jsou zde konektory pro připojení LCD displeje, ovládacích prvků, USB sběrnice a ICSP rozhranní. Zbylá oblast (na obr. 41 označená modře) je analogová část. Ta se skládá ze tří shodných bloků. Každý blok slouží k obsluze jedné osy posuvu. Součástí takovéhoto bloku je dvojce spínacích vysokonapěťových tranzistorů, relé sloužící k přepnutí polarity výstupního napětí a dva precizní 25-ti otáčkové trimry sloužící k nastavení velikosti kroku a maximální hodnoty výstupního napětí.

Obr. 42: Celkový pohled

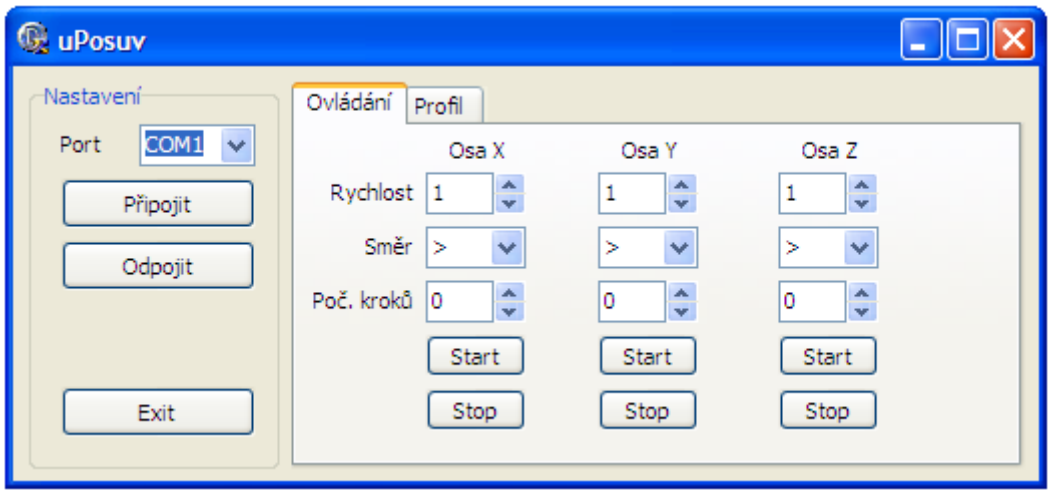
3.5 Program pro počítač

Pro pohodlnější ovládání mikroposuvu byl vytvořen program pro osobní počítač. Program byl napsán pomocí integrovaného vývojového prostředí CodeGear RAD Studio v jazyce Object Pascal. Protože využívá Microsoft .NET Framework, je nutné, aby na počítači, na kterém je spouštěn, byl tento framework nainstalován.

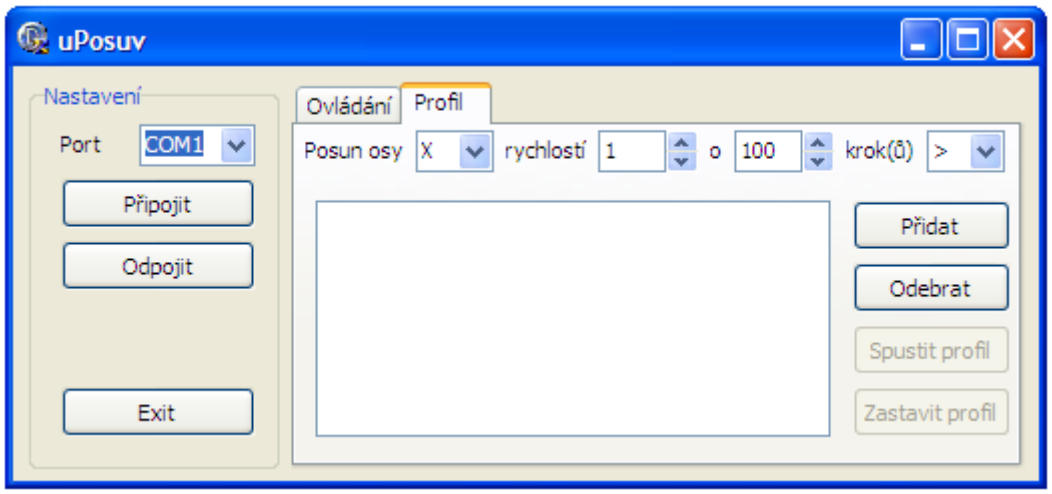
Na obr. 43 a 44 je uživatelské rozhranní programu ovládajícího mikroposuv. V položce „Nastavení“ vybereme port, po kterém bude probíhat komunikace. Po stisku tlačítka „Připojit“ se daný port otevře a začne probíhat komunikace.

V pravé části okna programu jsou dvě záložky pomocí kterých vybereme, jakým způsobem chceme zařízení ovládat. Pokud jsme na záložce „Ovládání“, nastavujeme jednotlivé parametry posuvu (rychlost, směr, počet kroků). Dále můžeme použít tlačítka „Start“ a „Stop“ ke spuštění, resp. zastavení posuvu. Po přepnutí na záložku „Profil“ můžeme definovat uživatelský profil. Pomocí tlačítka „Přidat“ vložíme do seznamu jednotlivé kroky

definovaného profilu. Stisknutím tlačítka „Odebrat“ můžeme ze seznamu vymazat aktuálně vybraný krok profilu. Po stisku tlačítka „Spustit profil“ je do zařízení přeneseno nastavení pro první krok profilu. Jakmile je tento krok dokončen, zařízení si vyžádá nastavení dalšího kroku. Takovýto postup se opakuje dokud nedojde k přenesení a vykonání posledního kroku profilu. Stisknutím tlačítka „Zastavit profil“ lze zastavit pohyb posuvu.



Obr. 43: Ovládací program – položka ovládání

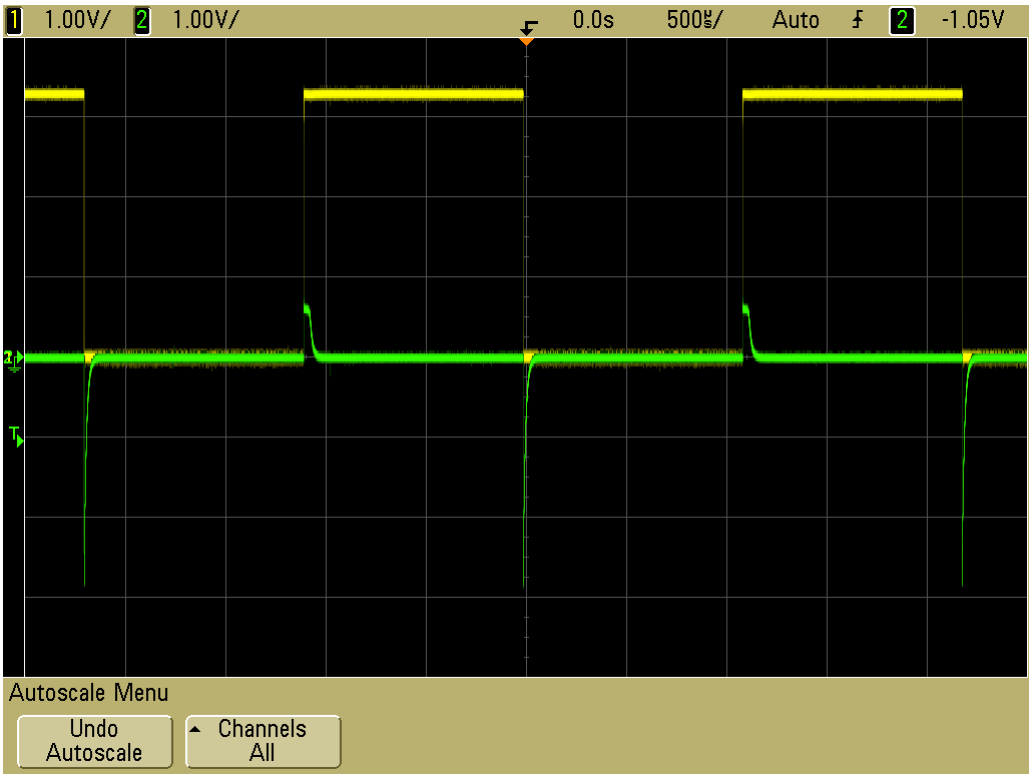


Obr. 44: Ovládací program – položka profil

4 Naměřené hodnoty

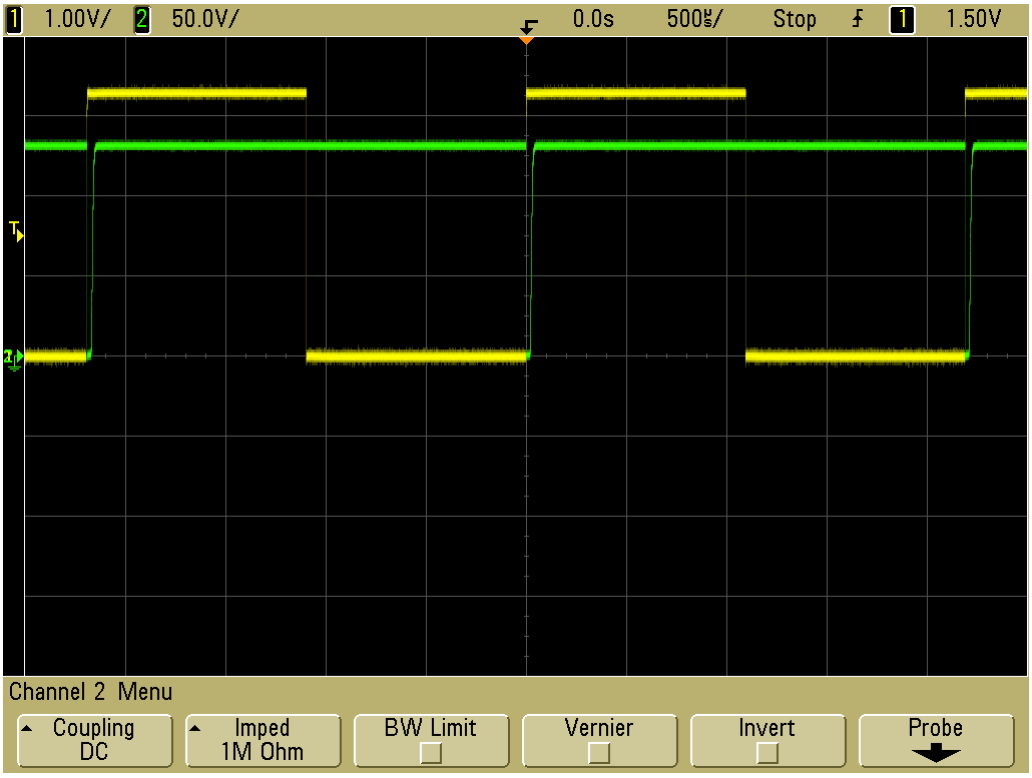
V této kapitole jsou porovnány naměřené hodnoty s teoretickými hodnotami simulací provedených v programu Orcad PSpice. Nasimulované průběhy napětí v jednotlivých uzlech analogového bloku jsou uvedeny v kapitole 4.2.2. na obrázku 17. Schéma zapojení tohoto bloku je na obrázku 16.

Vstupní signál analogové části U_{IN} (průběh generovaný mikrokontrolerem) je na obrázcích 45, 46 a 47 vyznačen žlutou čarou. Zelenou čarou je na obr. 45 vyznačen průběh napětí na bázi tranzistoru T_1. Na obr. 46 je vyznačen průběh napětí na kolektoru tranzistoru T2. A obrázek 47 zachycuje průběh výstupního napětí. Z uvedených průběhů plyne, že se naměřené hodnoty shodují s hodnotami teoretickými.

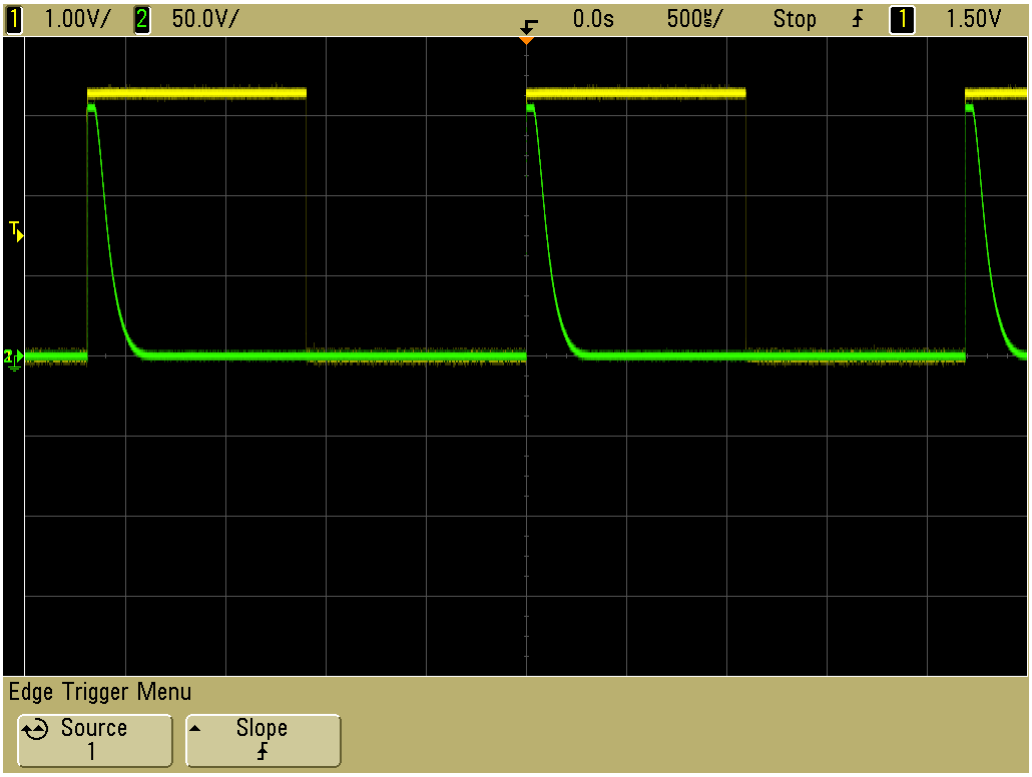


Obr. 45: Průběh napětí U_{IN} a U_1

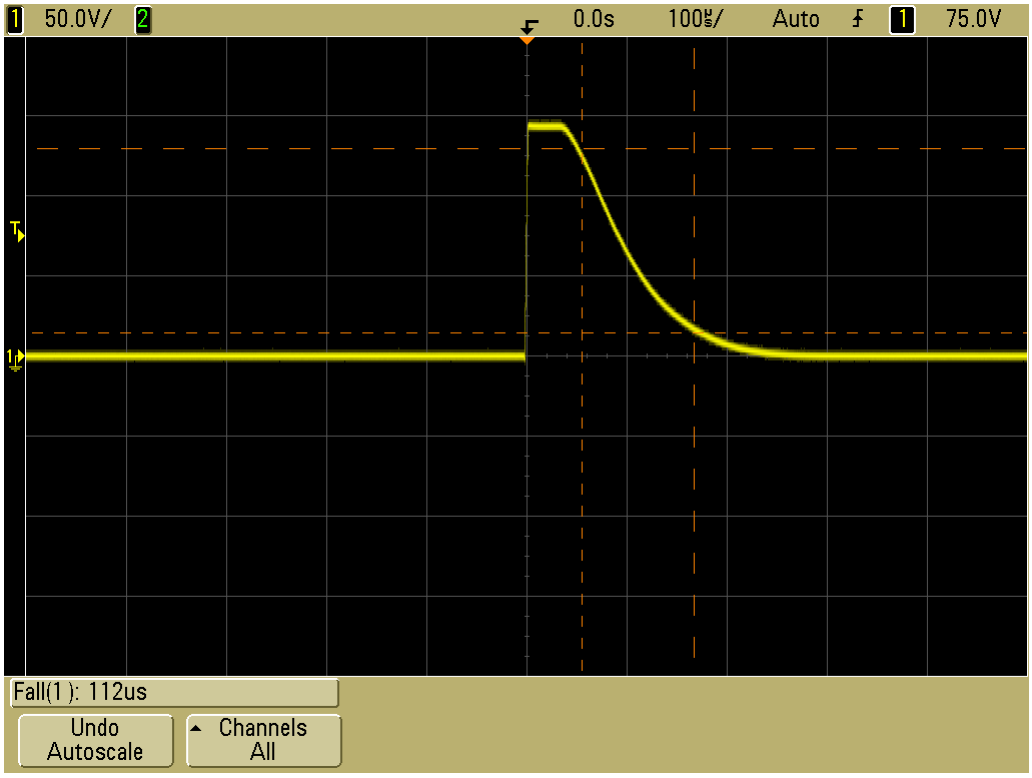
Na obrázcích 48 a 49 je znázorněn vliv odporu R_{C2} na délku impulsu. Čím větší bude hodnota tohoto odporu, tím déle se bude vybíjet kapacita tranzistoru T_2, a tím bude impuls delší.



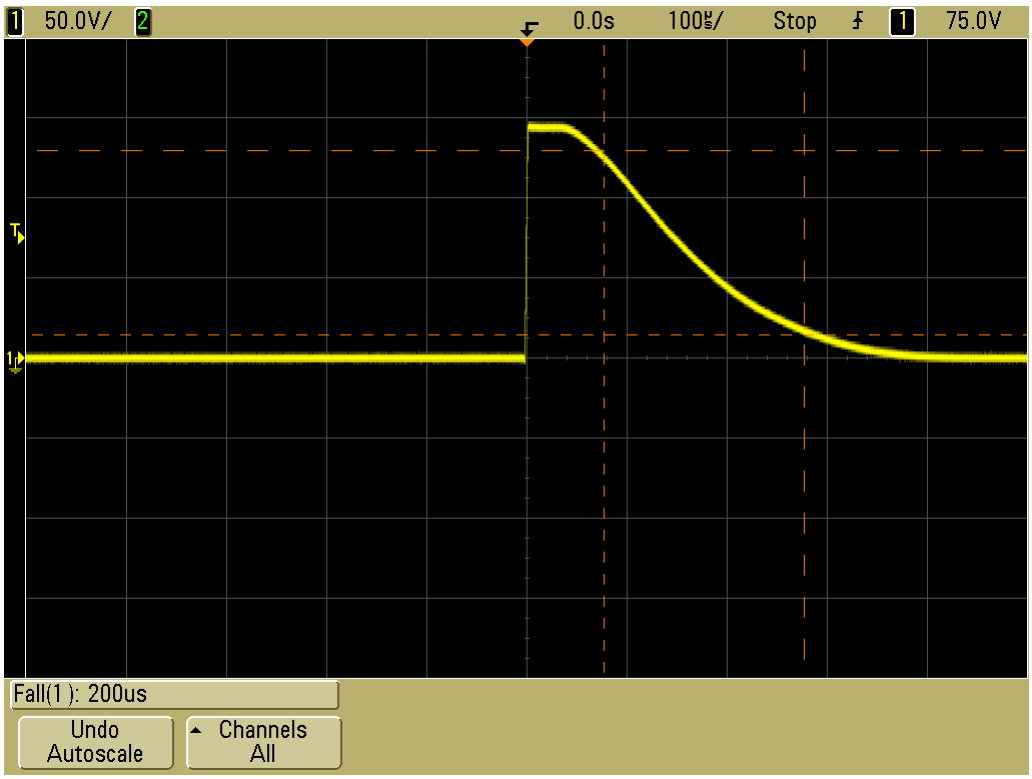
Obr. 46: Průběh napětí U_{IN} a U_2



Obr. 47: Průběh napětí U_{IN} a U_{OUT}



Obr. 48: Vliv odporu R_{C2} na délku pulsu, R_{C2}=330 kΩ



Obr. 49: Vliv odporu R_{C2} na délku pulsu, R_{C2}=830 kΩ

5 Závěr

Na tomto místě bych rád shrnul výsledky své práce, v níž jsem se zabýval návrhem a realizací zařízení, umožňující ovládání mikroposuvu. Na základě popisu mikroposuvu využívajícího obrácený piezoelektrický jev, o němž jsem psal v první části své práce, jsem následně provedl teoretický návrh takového zařízení.

Uvedený návrh jsem nejdříve podrobil počítačovým simulacím v programu Orcad PSpice. Výsledné hodnoty simulací, které uvádím v kapitole 3.2.2 na obrázku 17, odpovídají požadovanému průběhu signálu ovládajícího mikroposuv. Provedené simulace sloužily jako podklad pro realizaci teoretického návrhu. Zařízení ovládající mikroposuv jsem zhotovil a následně ověřil jeho funkci sérií měření. Naměřené hodnoty jsou uvedeny v kapitole 4 a prakticky se shodují s výsledky simulací. Zařízení prošlo také testovacím provozem, kde prokázalo svou schopnost ovládat mikroposuv.

Součástí zadání bylo také zhotovení programového vybavení pro osobní počítač. Vytvořil jsem tedy program umožňující nastavení jednotlivých parametrů posuvu, včetně možnosti přednastavovat uživatelsky definovaný profil posouvání.

Konstrukcí zařízení umožňující ovládat mikroposuv s nastavitelnou rychlostí posuvu a velikostí kroku a vytvořením počítačového programu na jeho ovládání bylo splněno zadání mé diplomové práce.

Na závěr předkládám několik námětů na další možné vylepšení. V současné verzi se velikost kroku nastavuje pomocí precizního trimru, jisté vylepšení by spočívalo v možnosti měnit velikost kroku elektronicky pomocí mikrokontroleru. Sejným způsobem by se mohlo řešit také nastavení maximální hodnoty výstupního napětí. Tyto parametry však po prvotním nastavení obvykle zůstávají konstantní, proto jejich elektronické ovládání není nezbytností. Z důvodu mechanického opotřebení kontaktů relé, přepínajícího polaritu výstupního signálu, by bylo případně možné jej nahradit zapojením tranzistorů do tzv. H – můstku.

6 Použitá literatura

[1] KMÍNEK, M., KADLEC, K. *Měřící a řídící technika*. 2000. Dostupné z WWW: http://uprt.vscht.cz/ucebnice/mrt/F0-ram.htm

[2] USB-IF. *Universal Serial Bus Specification Revision 2.0*. 2000. 650 s. Dostupné z WWW: http://www.usb.org/developers/docs/

[3] NATIONAL SEMICONDUCTOR. Santa Clara, Kalifornia. *LM78M05 - 3-Terminal Positive Voltage Regulator*. 2005. 11 s.
Dostupné z WWW: http://www.national.com/pf/LM/LM78M05.html

[4] TEXAS INSTRUMENTS. Dallas, Texas. *REG103 - DMOS 500mA Low-Dropout Regulator*. 2005. 26 s.
Dostupné z WWW: http://focus.ti.com/docs/prod/folders/print/reg103-5.html

[5] SEME LAB. Leicestershire, UK. *2N3439 – High voltage NPN transistor*. 2002. 2 s. Dostupné z WWW: http://www.semelab.co.uk/pdf/bipolar/2N3439_40.pdf

[6] STMICROELECTRONICS. Geneva, Switzerland. *2N5416 – Silicon PNP transistor*. 2000. 4 s.
Dostupné z WWW: http://www.st.com/stonline/products/literature/ds/5233.htm

[7] MICROCHIP TECHNOLOGY INC. Chandler, Arizona. *PIC18F2455/2550/4455/4550 Data Sheet*. 2007. 430 s. Dostupné z WWW:
http://www.microchip.com/stellent/idcplg?IdcService=SS_GET_PAGE&nodeId=1335&dDocName=en010300

[8] ELECTRONIC ASSEMBLY GMBH. Gräfelfing, Germany. *EA W128-6X8/-6X9 Grafik 128x64 mit kontroller KS0713*. 2006. 12 s. Dostupné z WWW: http://www.lcd-module.de/eng/pdf/grafik/w128-6xx.pdf

[9] SAMSUNG SEMICONDUCTOR EUROPE GMBH. Eschborn, Germany. *KS0713 - 65 COM / 132 SEG Driver & controller for STN LCD*. 2000. 67 s. Dostupné z WWW: http://www.lcd-module.de/eng/pdf/zubehoer/ks0713.pdf

[10] BRESENHAM, J, *Algorithm for Computer Control of a Digital Plotter*, IBM Systems Journal, 4(1):25-30, 1965.

[11] GM ELECTRONIC. Praha, ČR. *P-RE30S – Rotační kodér s mechanickým kontaktem*. 2500. 1 s. Dostupné z WWW: http://www.gme.cz/_dokumentace/dokumenty/532/532-087/dsh.532-087.1.pdf

[12] MICROCHIP TECHNOLOGY INC. Chandler, Arizona. *MCHPFSUSB Firmware Userís Guide*. 2007. 28 s. Dostupné z WWW: http://www.microchip.com/USB

[13] USB-IF. *Universal Serial Bus Class Definitions for Communication Devices*. 1999. 121 s. Dostupné z WWW: http://www.usb.org/developers/devclass_docs/usbcdc11.pdf

[14] ROJVANIT, R. *Migrating applications to USB from RS-232 UART with minimal impact on software*. 2004. 16 s. Dostupné z WWW: http://www.microchip.com/stellent/idcplg?IdcService=SS_GET_PAGE&nodeId=1824&appnote=en021631

[15] HOLEČEK, O. *CRC (kontrolní součet)*. 2003 Dostupné z WWW: http://www.root.cz/clanky/crc-kontrolni-soucet/

[16] MICROCHIP TECHNOLOGY INC. Chandler, Arizona. *MPLAB Integrated Development Environment User's Guide*. 2006. 288 s. Dostupné z WWW: http://www.microchip.com/stellent/idcplg?IdcService=SS_GET_PAGE&nodeId=1406&dDocName=en019469

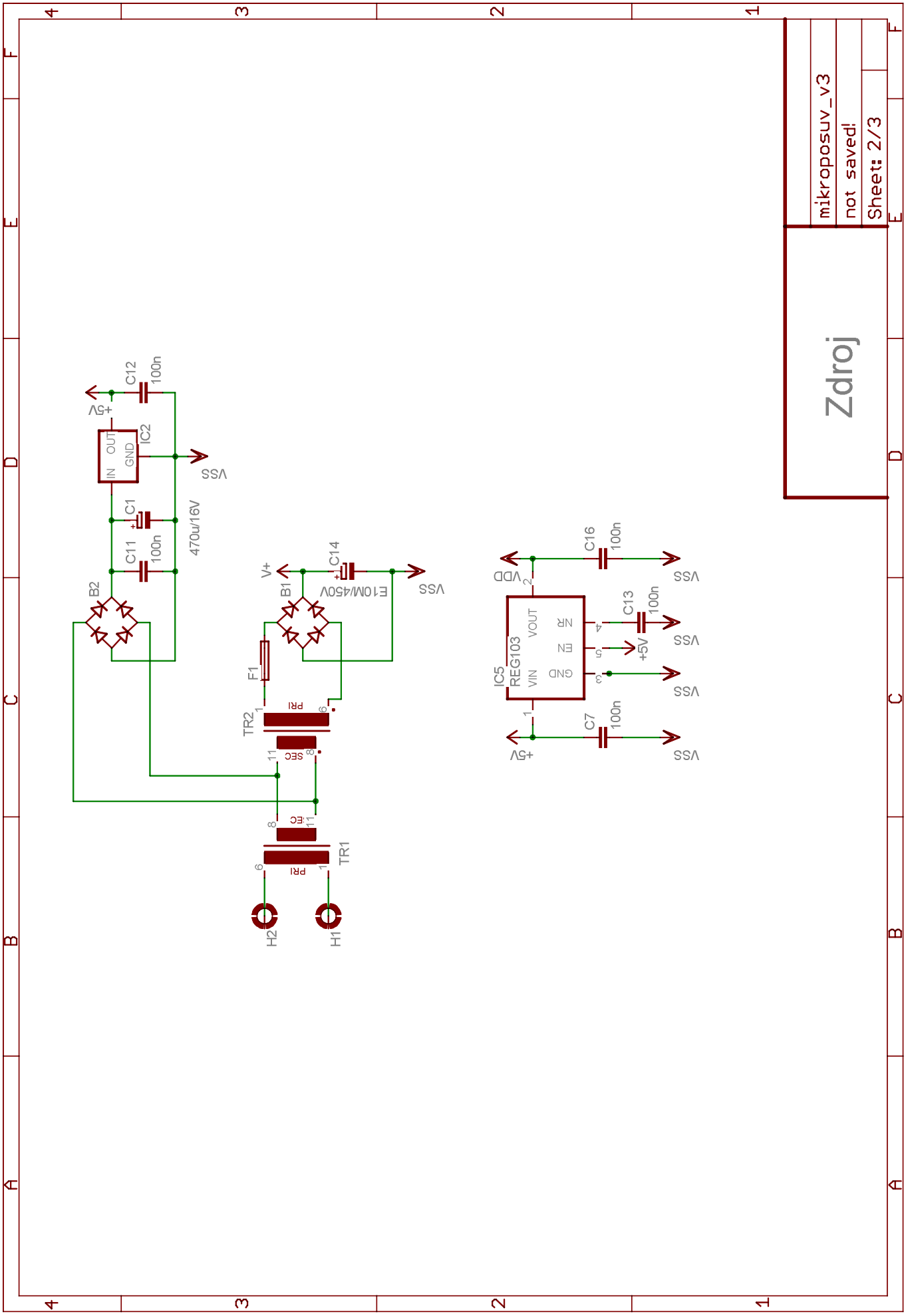
[17] MICROCHIP TECHNOLOGY INC. Chandler, Arizona. *MPLAB C18 C Compiler user's guide*. 2005. 136 s. Dostupné z WWW: http://www.microchip.com/stellent/idcplg?IdcService=SS_GET_PAGE&nodeId=1406&dDocName=en010014

7 Přílohy

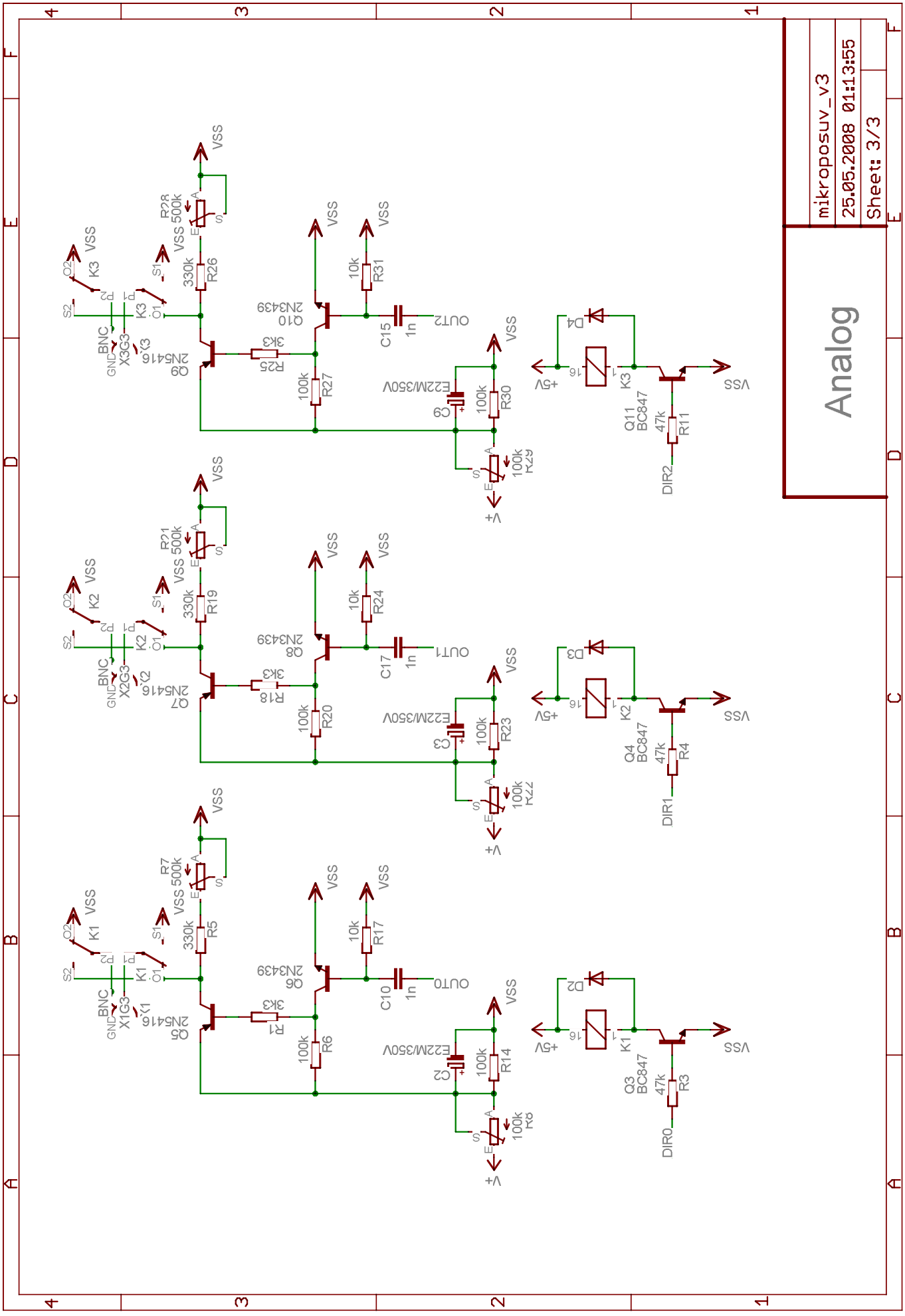
7.1 Obsah přiloženého CD

- Text diplomové práce
- Katalogové listy použitých obvodů
- Zdrojové kódy programu pro mikrokontroler
- Zdrojové kódy ovládacího programu
- Schéma a DPS ve formátu EAGLE

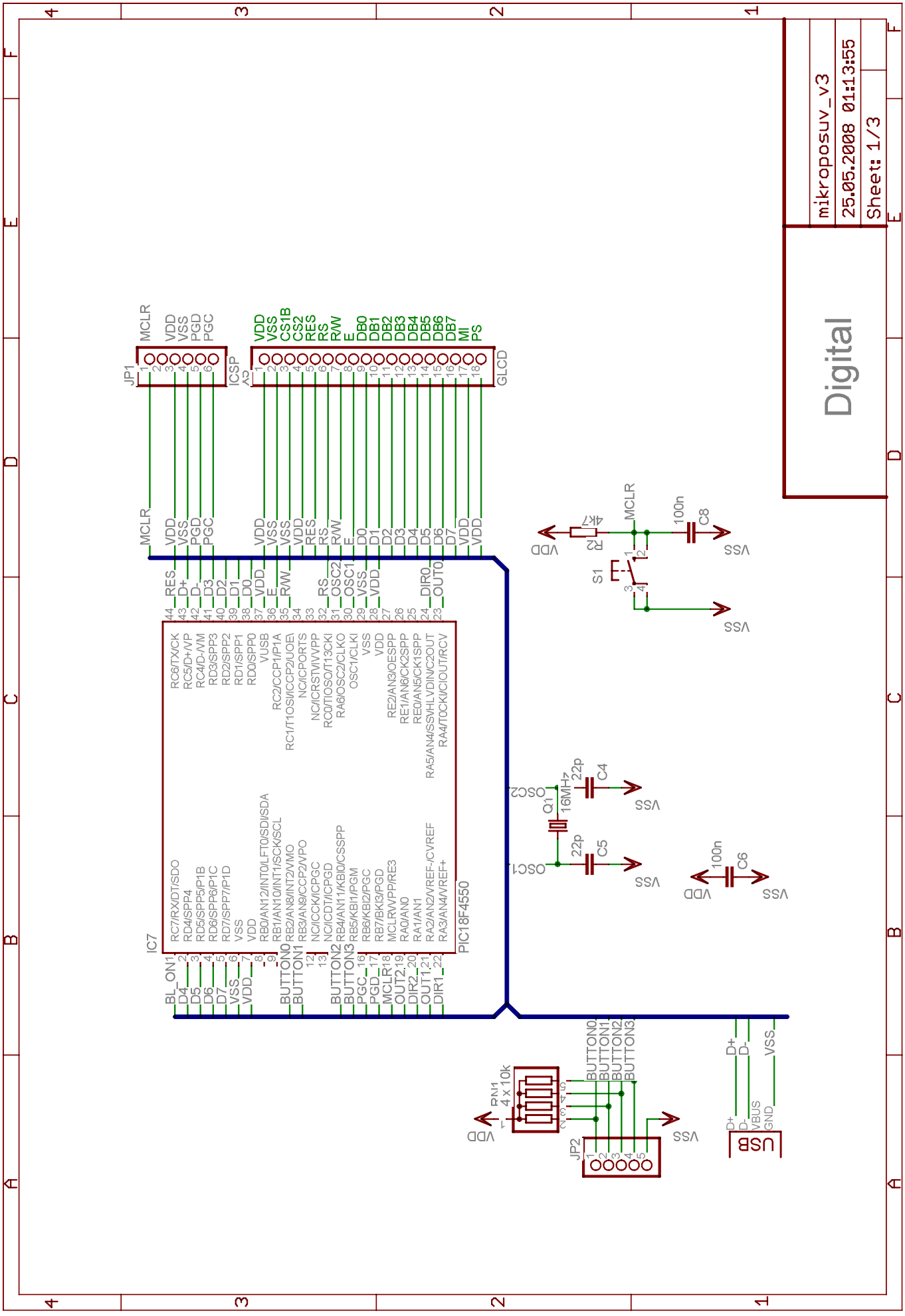
7.2 Schéma zapojení - zdrojová část



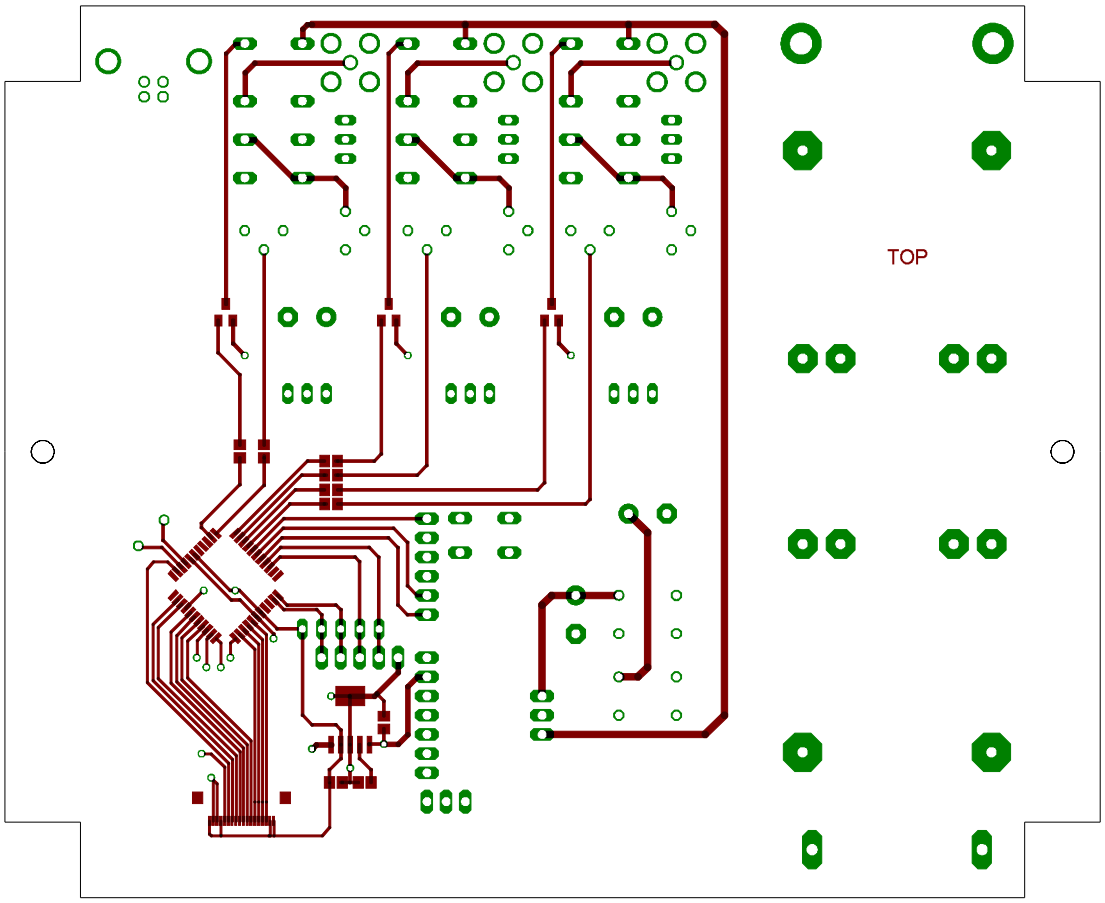
7.2 Schéma zapojení - analogová část



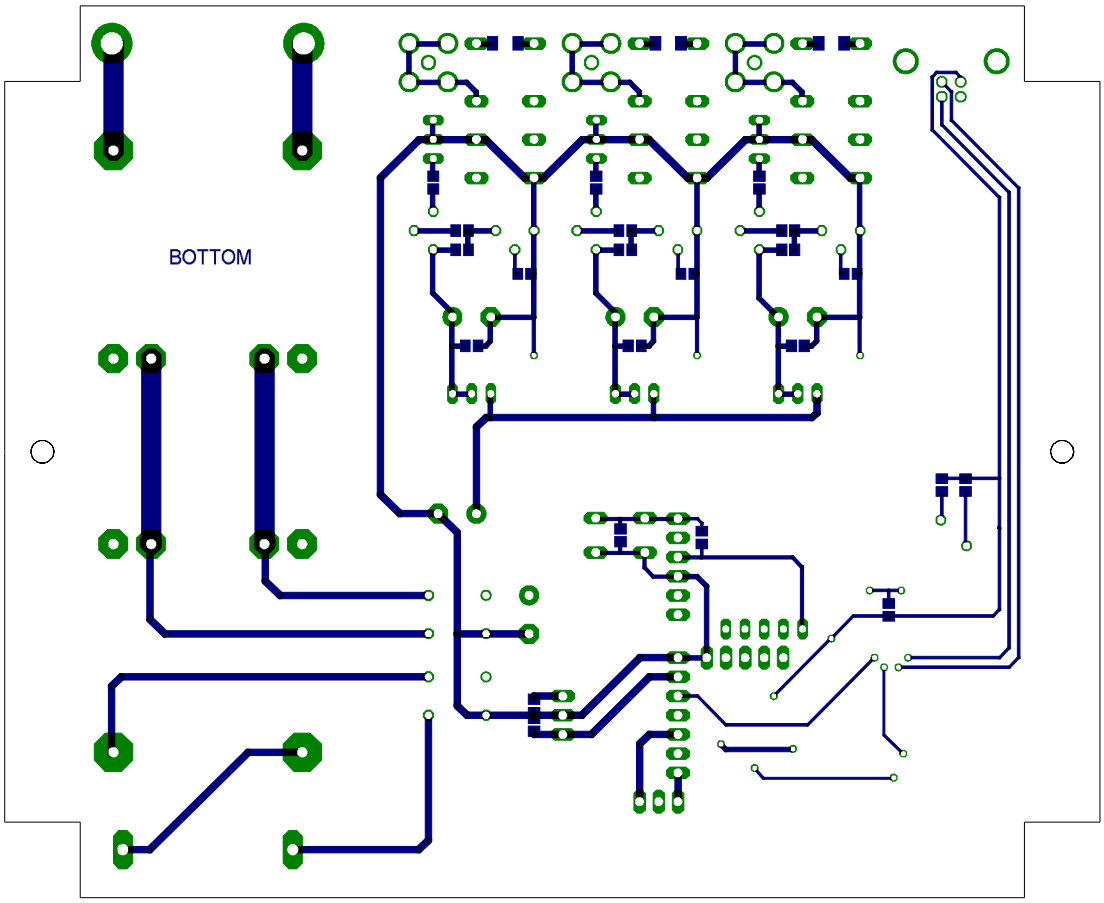
7.2 Schéma zapojení - digitální část



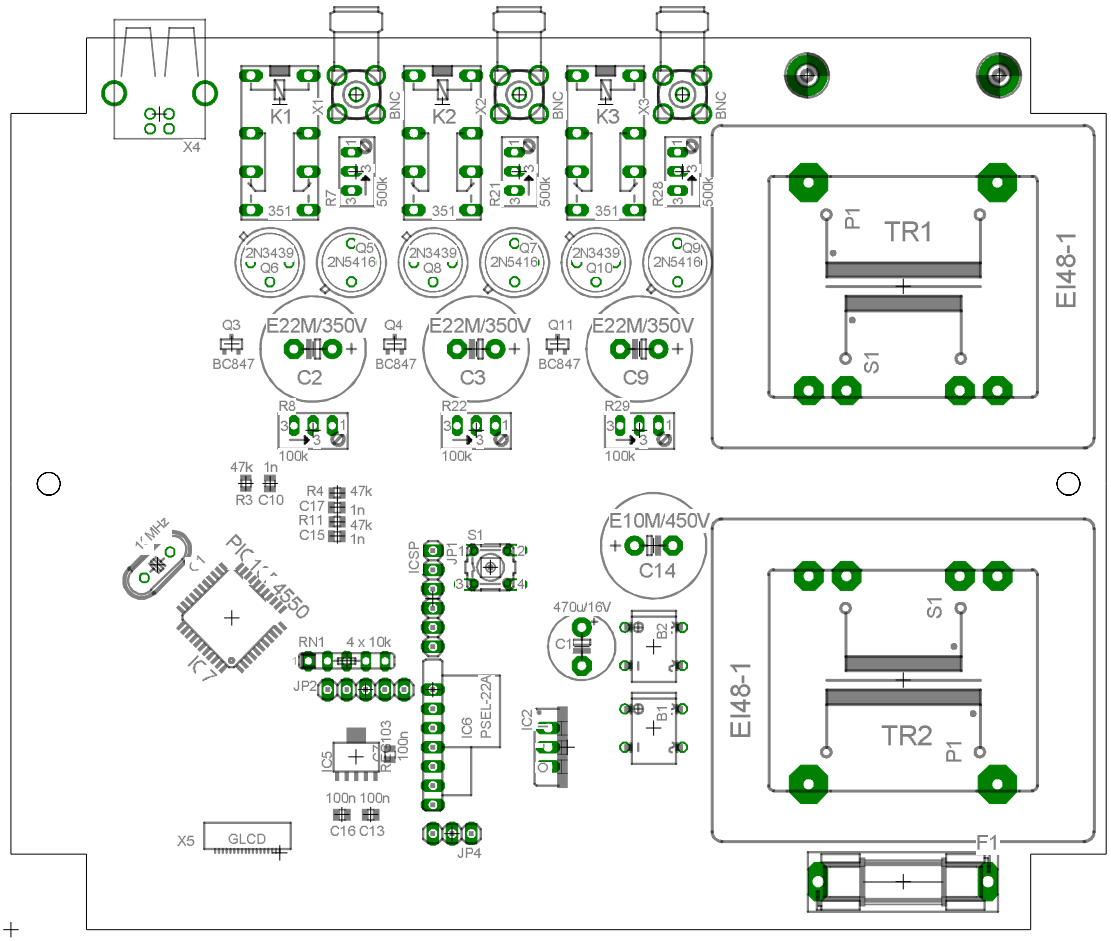
7.3 DPS - horní vrstva



7.3 DPS - spodní vrstva



7.4 Osazovací výkres - horní strana



7.4 Osazovací výkres - spodní strana